National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

ELECTRIC HEATING OF OIL SAND AND SIMILAR MATERIALS USING MULTIPHASE EXCITATION

BY ©

KERRY VEKVED

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE.

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

FALL 1992

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-77086-4

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: KERRY VEKVED

TITLE OF THESIS: ELECTRIC HEATING OF OIL SAND AND SIMILAR
MATERIALS USING MULTIPHASE EXCITATION

DEGREE: MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED: 1992

PERMISSION IS HEREBY GRANTED TO THE UNIVERSITY OF ALBERTA LIBRARY TO REPRODUCE SINGLE COPIES OF THIS THESIS AND TO LEND OR SELL SUCH COPIES FOR PRIVATE, SCHOLARLY OR SCIENTIFIC RESEARCH PURPOSES ONLY.

THE AUTHOR RESERVES ALL OTHER PUBLICATION AND OTHER RIGHTS IN ASSOCIATION WITH THE COPYRIGHT IN THE THESIS, AND EXCEPT AS HEREINBEFORE PROVIDED NEITHER THE THESIS NOR ANY SUBSTANTIAL PORTION THEREOF MAY BE PRINTED OR OTHERWISE REPRODUCED IN ANY MATERIAL FORM WHATEVER WITHOUT THE AUTHOR'S PRIOR WRITTEN PERMISSION.

Kerry Vekved

RR 1 Sexsmith
Alberta, Canada
TOH 3CO

Date: SEPT. 14, 1992

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

THE UNDERSIGNED CERTIFY THAT THEY HAVE READ, AND RECOMMEND TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH FOR ACCEPTANCE, A THESIS ENTITLED

ELECTRIC HEATING OF OIL SAND AND SIMILAR MATERIALS USING MULTIPHASE EXCITATION

SUBMITTED BY KERRY VEKVED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE.

F Chute
F. S. Chute

FE Vermeulen
F. E. Vermeulen

A. Craggs.
A. Craggs

JF Vaneldik
J. F. Vaneldik

Date: JULY 28, 1992

Abstract

The high viscosity of oil in oil sand and heavy oil formations can be effectively reduced by heating with low frequency electromagnetic energy. A sufficiently lowered viscosity will enable the oil to flow, allowing more effective recovery than otherwise possible.

Embedded arrays of vertical electrodes are typically used to deliver the electromagnetic energy to the underground formations. In the past, implementation schemes have called for single phase excitation of these electrodes. However, it is often difficult to generate the desired degree of uniformity when heating large volumes of oil sand with this form of excitation. To attain more uniform heating, it has been suggested that electrodes be excited with several different phases.

This thesis investigates multiple phase excitation. The investigation was conducted by using a numerical simulator based on a finite difference approximation of Maxwell's equations and the heat equation. With the simulator, multiphase excitation of several different electrode arrays was modelled. In addition to array patterns that have been proposed for heating large volumes of oil sand, circular arrays and multiple electrode configurations for a single well were also studied.

Results of these simulations revealed that, in general, multiphase excitation can be used to generate a more even distribution of heat around a given array than single phase excitation. The reason for this is that multiphase excitation generates a greater multiplicity of current paths through the formation than single phase excitation. In addition, it was found that alteration of the phase separation between electrodes can be used to tailor, somewhat, the heating pattern achieved in the formation.

Acknowledgements

There are several people to whom which I would like to convey my appreciation for their part in helping me complete this thesis. They are:

Dr. Chute & Dr. Vermeulen - for supervising this project and for providing financial support when I had no other sources of income.

Dr. Stroemich - for providing constructive criticism in the preparation of this thesis and for sharing knowledge gained from his experience.

Dr. Sumbar - for sharing his knowledge of the Macintosh system and for helping me solve many of the software problems that I encountered.

Dr. Lawson - for providing guidance and a little extra rope.

Scott Juskiw - for the programming and troubleshooting that he did for me as a representative of the Apple Research Partnership Program.

Jim Fearn - for his thought provoking practicality.

I would also like to especially thank the Alberta Oil Sands Technology and Research Authority for the financial support which they provided.

Table of Contents

Acknowledgements i

List of Tables iii

List of Figures iv

List of Symbols x

Chapter

1. Background to the Study of Multiphase Excitation 1
    - Introduction 1
    - Present Investigation 8
    - Theory 10
2. Details of the Investigation 15
3. Results of the Computer Simulations 26
    - The Four-Spot Array 26
    - The Five-Spot Array 37
    - The Seven-Spot Array 46
    - The Circular Array 54
    - The Single Well Configuration 65
    - Power Losses 88
4. Interpretation of the Results 96
    - Summary and Conclusions 96
    - Recommendations 99

References 101

Appendices

A. Changes Made to EPEIOS in Transferring It From the Amdahl to the Macintosh Computer 104

B. Developmental Changes Made to the EPEIOS code 108

C. EPEIOS User's Manual 156

Tables

Table Page

3.1 Electrode Impedances in an Isotropic Medium for Various Three Electrode Configurations With Three Phase Excitation. 81

Figures

Figure Page

2.1 Approximation of No-Flow Boundary Diagonal to Grid Orientation. 20

3.1 Large Array Formed From Repeated Four-Spot. 27

3.2 (a) Plan View of Modelled Four-Spot Subarray with Polarities and Current Pathways Shown for the First Single-Phase Excitation Scheme. 28

3.2 (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Four-Spot Subarray with the Single-Phase Excitation Shown in (a). Total Input Power Is 50 kW. 28

3.3 (a) Plan View of Modelled Four-Spot Subarray with Polarities and Current Pathways Shown for the Second Single-Phase Excitation Scheme. 32

3.3 (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Four-Spot Subarray with the Single-Phase Excitation Shown in (a). Total Input Power Is 50 kW. 32

3.4 (a) Plan View of Modelled Four-Spot Subarray with Phase Angles and Current Pathways Shown for Four-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. 35

3.4 (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Four-Spot Subarray with Four-Phase Excitation. Total Input Power Is 50 kW. 35

3.5 Large Array Formed From Repeated Five-Spot Pattern. 38

3.6 (a) Plan View of Modelled Five-Spot Subarray with Polarities and Current Pathways Shown for Single-Phase Excitation. 39

3.6 (b) Heating Rates ($W\cdot m^{-3}$) in Horizontal (x-y) Plane for Five-Spot Subarray with Single-Phase Excitation. Total Input Power Is 50 kW. 39

3.7 (a) Plan View of Modelled Five-Spot Subarray with Phase Angles and Current Pathways Shown for Standard Three-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. 41

3.7 (b) Heating Rates [$W\cdot m^{-3}$] in Horizontal (x-y) Plane for Five-Spot Subarray with Three-Phase Excitation as Shown in (a). Total Input Power Is 50 kW. 41

3.8 (a) Heating Rates Normalized to 1.0×10^4 $W\cdot m^{-3}$ Along the Plane y=5 m for Single-Phase Excitation of the Five-Spot Subarray. 43

3.8 (b) Heating Rates Normalized to 1.0×10^4 $W\cdot m^{-3}$ Along the Plane y=5 m for Three-Phase Excitation of the Five-Spot Subarray. 43

3.9 (a) Plan View of Modelled Five-Spot Subarray with Phase Angles and Current Pathways Shown for Three-Phase Excitation That Has Unequal Separation Between Phases. Phase Angles Are Given with Respect to a Common Reference. 45

3.9 (b) Heating Rates [$W\cdot m^{-3}$] in Horizontal (x-y) Plane for Five-Spot Subarray with Three-Phase Excitation as Shown in (a). Total Input Power Is 50 kW. 45

3.10 Large Array Formed From Repeated Seven-Spot Pattern. 47

3.11 (a) Plan View of Seven-Spot Array with Polarities and Current Pathways Shown for Single-Phase Excitation. 49

3.11 (b) Heating Rates [$W\cdot m^{-3}$] in Horizontal (x-y) Plane for Single-Phase Excitation of the Seven-Spot Array in a

Problem Domain That Is Seemingly Infinite in the Horizontal Plane. Total Input Power Is 100 kW. 49

3.12 (a) Heating Rates Normalized to 1.0×10^4 W·m^{-3} Along the Plane y=18.7 m for Single-Phase Excitation of the Seven-Spot Array. 51

3.12 (b) Heating Rates Normalized to 1.0×10^4 W·m^{-3} Along the Plane y=18.7 m for Three-Phase Excitation of the Seven-Spot Array. 51

3.13 (a) Plan View of Modelled Seven-Spot Array with Phase Angles and Current Pathways Shown for Three-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. 53

3.13 (b) Heating Rates (W·m^{-3}) in Horizontal (x-y) Plane for Three-Phase Excitation of the Seven-Spot Array in a Problem Domain That Is Seemingly Infinite in the Horizontal Plane. Total Input Power Is 100 kW. 53

3.14 (a) Plan View of Modelled Hexagonal Array with Phase Angles and Current Pathways Shown for Six-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. 56

3.14 (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Hexagonal Array with Six-Phase Excitation as Shown in (a). Total Input Power Is 50 W. 56

3.15 (a) Plan View of Modelled Hexagonal Array with Polarities and Current Pathways Shown for Single-Phase Excitation. 59

3.15 (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Hexagonal Array with Single-Phase Excitation as Shown in (a). Total Input Power Is 50 W. 59

3.16 (a) Plan View of Modelled Octagonal Array with Phase

Angles and Current Pathways Shown for Eight-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. 61

3.16 (b) Heating Rates [$W \cdot m^{-3}$] in Horizontal (x-y) Plane for Octagonal Array with Eight-Phase Excitation as Shown in (a). Total Input Power Is 50 W. 61

3.17 (a) Plan View of Modelled Dodecagonal Array with Phase Angles and Current Pathways Shown for Twelve-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. 62

3.17 (b) Heating Rates [$W \cdot m^{-3}$] in Horizontal (x-y) Plane for Dodecagonal Array with Twelve-Phase Excitation as Shown in (a). Total Input Power Is 50 W. 62

3.18 Heating Rates for the Different Circular Arrays Taken Along the Vertical (y-z) Plane Which Passes Through the Center of the Array. Heating Rate Values Are Normalized to 2×10^3 $W \cdot m^{-3}$. 64

3.19 Plan View of an Array of Single Wells and Corresponding Planes of Symmetry. 66

3.20 Problem Domain for Top/Bottom Configuration. 69

3.21 Heating Rates [$W \cdot m^{-3}$] in Vertical (y-z) Plane for Single-Phase Excitation of Top/Bottom Configuration Shown in Figure 3.20. Formation is Layered (Interface at z = 32.5 m) with Electrical Conductivity Approximately Ten Times Greater In Top Layer Than in Bottom Layer. Total Input Power Was 40 kW. 70

3.22 Temperatures [°C] in Vertical (y-z) Plane for Single-Phase Excitation of Top/Bottom Configuration Shown in Figure 3.20 After Heating From an Initial Temp. of 10°C to a Maximum Formation Temp. of 150°C. Formation Is Layered (Interface at z = 32.5 m) with

Electrical Conductivity in Top Layer Approximately Ten Times That in the Bottom Layer. Input Power Was a Constant 40 kW and Total Heating Time Was 174.5 Hours 73

3.23 (a) Three Electrode Configuration for 20 m Well Section with Electrodes 1 m Long and Insulated Sections 8.5 m Long. Phase Angles with Respect to a Common Reference Are Shown for Three-Phase Excitation. 76

3.23 (b) Heating Rates [W·m^{-3}] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW. 76

3.24 (a) Three Electrode Configuration for 20 m Well Section with Electrodes 4 m Long and Insulated Sections 4 m Long. Phase Angles with Respect to a Common Reference Are Shown for Three-Phase Excitation. 77

3.24 (b) Heating Rates [W·m^{-3}] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW. 77

3.25 (a) Three Electrode Configuration for 20 m Well Section with Electrodes 6.4 m Long and Insulated Sections 0.4 m Long. Phase Angles with Respect to a Common Reference Are Shown for Three-Phase Excitation. 79

3.25 (b) Heating Rates [W·m^{-3}] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW. 79

3.26 (a) Three Electrode Configuration for 20 m Well Section with Center Electrode 4 m Long, End Electrodes 5 m Long and Insulated Sections 3 m Long. Phase Angles with Respect to a Common Reference Are

Shown for Three-Phase Excitation. 83

3.26 (b) Heating Rates [$W \cdot m^{-3}$] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW. 83

3.27 Temperatures [°C] in Vertical (y-z) Plane for Three-Electrode Configuration Shown in Figure 3.26(a) in an Isotropic Medium After Heating for 174.5 Hours. Power Input Was a Constant 40 kW. 85

3.28 Temperatures [°C] in Vertical (y-z) Plane for Three-Electrode Configuration Shown in Figure 3.26(a) in a Layered Formation (Interface at z = 32.5 m) After Heating for 174.5 Hours. Electrical Conductivity In the Top Layer Was Approximately Ten Times That In the Bottom Layer. Input Power Was a Constant 40 kW. Initial Temperature Was 10 °C and Maximum Final Temperature Was 133 °C. 87

Symbols

Symbol	Quantity	Units
a	radius of casing	[m]
$\mathbf{B}$	magnetic flux density	[T]
$\mathbf{D}$	electric flux density	$[\mathrm{C{\cdot}m^{-2}}]$
$\mathbf{E}$	electric field intensity	$[\mathrm{V{\cdot}m^{-1}}]$
$\mathbf{H}$	magnetic field intensity	$[\mathrm{A{\cdot}m^{-1}}]$
I_L	line current	[A]
$\mathbf{J}$	electric current density	$[\mathrm{A{\cdot}m^{-2}}]$
j	$\sqrt{-1}$	
k	thermal conductivity	$[\mathrm{W{\cdot}m^{-1}{\cdot}K^{-1}}]$
l	length of casing	[m]
M	volumetric heat capacity	$[\mathrm{J{\cdot}m^{-3}{\cdot}K^{-1}}]$
$P_{3\emptyset}$	power lost in three phase cables	[W]
$P_{top/bottom}$	power lost in top/bottom configuration	[W]
Q	heating rate	$[\mathrm{W{\cdot}m^{-3}}]$
R_{cable}	resitance of three phase cable	$[\Omega]$
R_{casing}	internal resistance of well casing	$[\Omega]$
$R_{electrode}$	electrode resistance	$[\Omega]$
R_{rod}	electrode impedance of well casing	$[\Omega]$
R_{tubing}	internal resistance of production tubing	$[\Omega]$
T	temperature	[°C]
Z_{tubing}	internal impedance of production tubing	$[\Omega]$
Z_{casing}	internal impedance of well casing	$[\Omega]$
ε	permittivity	$[\mathrm{F{\cdot}m^{-1}}]$

$\eta_{top/bottom}$	efficiency of power delivery (top/bottom)	[%]
$\eta_{3\phi}$	efficiency of power delivery (3 phase)	[%]
σ	electrical conductivity	[$S \cdot m^{-1}$]
Φ	electric scalar potential	[V]
ω	angular frequency	[$rad \cdot s^{-1}$]

CHAPTER 1

Background to the Study of Multiphase Excitation

Introduction

Nearly one trillion barrels of oil are estimated to reside in oil sand deposits in the province of Alberta in Canada (Strom & Dunbar, 1979). To date, this valuable resource remains largely untapped because the oil possesses a relatively high viscosity which causes it to be essentially immobile in its natural state. Thus, recovery by the use of conventional drilling and pumping techniques is precluded. Currently, oil is being extracted from Alberta oil sand by surface mining techniques but, unfortunately, only about 10% of the total oil sand deposit is located close enough to the surface to make this a practical method of recovery (*AOSTRA*, 1990). As a result, a considerable amount of research has gone into developing viable *in situ* recovery techniques that would enable extraction of the vast amounts of oil located in the more deeply bedded oil sand formations.

The most common and seemingly the most practical approach to recovery of the oil has been to utilize some form of thermal recovery method (Carrigy, 1983). With such a method, heat would be transferred to the oil in formation, thereby increasing its temperature. A sufficient temperature rise would cause reduction in the viscosity of the oil to the point where it would be able to flow, perhaps with the aid of some form of reservoir flooding technique.

Conventional thermal recovery techniques, such as steam or hot water injection, combine heating with flooding, but are severely limited by the characteristic nature of deeply buried oil sand (Vermeulen & Chute, 1983). Typically, oil sand formations in Alberta are quite impermeable and thus do not allow deep penetration of injected steam or hot water. In addition, the relatively low thermal conductivity of oil sand impedes the flow of heat away from the injector site. These two factors, combined with the lack of directivity that is inherent to any underground injection scheme, pose serious obstacles to *in-situ* thermal recovery methods in oil sand formations.

The inefficacy of conventional injection techniques prompted research into discovering unconventional heating methods that would be able to quickly and efficiently heat large volumes typically found several hundreds of meters below the surface. One such method that has had success and continues to show promise is electromagnetic heating (Chute & Vermeulen, 1988; Glandt & Hsu, 1992). This technique is not limited by the low thermal conductivity or low permeability of oil sand as heating is effected from within the formation. Electromagnetic energy is delivered to the oil sand via electrode wells embedded in the formation and large volumes can be heated by using multiple electrodes in some form of array. Most of the electromagnetic energy is deposited in the volume between the electrodes. Thus, with electromagnetic heating, the zone to be heated can be more accurately selected than with conventional heating methods.

There are basically two mechanisms by which electromagnetic energy can produce heating in oil sand formations. One is heating due to ohmic losses and the other is dielectric heating. Heating due to ohmic losses dominates at low frequencies which for oil sand heating typically means less than about one megahertz. Dielectric heating dominates at high frequencies which are generally considered to be radio frequencies (greater than 1 MHz) and above (Vermeulen, Chute & Cervenan, 1979). As a result of the two distinct mechanisms involved in electromagnetic heating of oil sand, two different classes of heating techniques defined by excitation frequency have emerged.

Low frequency heating techniques or those that generate heat through ohmic losses, require a relatively high moisture content in the oil sand to be effective. The reason is that at low frequencies current flow in the formation is primarily by way of ionic conduction through the connate water. In this manner, the oil sand formation acts, in effect, as a large resistive element and thus ohmic losses occur. The larger oil sand deposits found near Athabasca in northern Alberta typically possess a sufficiently high moisture content so that low frequency heating can be used. The electromagnetic energy is able to penetrate deep into the formation thereby contributing to heating uniformity. With a source frequency of 60 Hz, depth of penetration is of the order of several hundred meters (Vermeulen & Chute, 1983). The major drawback of heating at low frequencies has been and continues to be that prolonged heating tends to raise the temperature of the water in the formation to the point of vaporization. This is especially true near the electrodes where current densities and consequently ohmic losses and heating rates are highest. Removal of moisture from the oil sand by vaporization effectively breaks the electrical circuit. The formation of a dry space of a few millimeters thickness around an electrode is sufficient to prevent the flow of current, thus causing the cessation of the heating process. Several methods have been proposed to combat overheating at the electrodes. These include artificially increasing the effective electrode diameter, pressurization of the formation near the electrode, circulation of cooling fluids near the electrode (Chute, Vermeulen & Stevens, 1987) and control of input power (Vermeulen & Chute, 1983).

Heating at high frequencies is not hampered by the potential formation of a moisture-free region surrounding an electrode since such a dry zone could be capacitively bridged by the current at these higher frequencies. As a result, oil sand formations can be heated to temperatures much higher than the vaporization temperature of water when heating at radio frequencies. However, the depth to which electromagnetic energy is able to penetrate into naturally occurring formations of Athabasca oil sand is generally restricted to a few meters (Vermeulen & Chute, 1983). The water in the formation will initially absorb most of the energy. Of course, propagation distances increase as moisture is removed by vaporization. Nevertheless, the limited penetration depths necessitate very close electrode spacing. This can lead to relatively high initial emplacement costs. In addition, such a system also requires frequency conversion equipment in order to provide a high frequency electric power source, which further adds to the initial cost. A radio frequency heating process using horizontal wells has been studied for Athabasca oil sand (McPherson, Chute & Vermeulen, 1986).

Because of its deeper penetration, heating with low frequency electromagnetic energy allows for greater separation distances between electrodes than when heating at radio frequencies. In addition, electrical power at 60 Hz is readily available as a low frequency source. These factors make 60 Hz presently the frequency of choice for heating Athabasca oil sand. Consequently, a good deal of research has gone into developing a workable electromagnetic heating system at this frequency.

Due to the reliance of the heating process at low frequencies on the presence of a continuous water path through the formation between the electrodes, temperatures in excess of the *in situ* vaporization temperature of water can not be achieved when heating at 60 Hz. As a result, a 60 Hz heating process is best suited as a formation preheat whereby electrical energy would be used to raise the temperature of the oil sand to the point that steam injection would become a practical and efficient thermal recovery technique. Chute, Vermeulen and Stevens (1987) have shown this combination to be economically viable.

Used in a preheat capacity, it is desirable for the electric heating system to be able to control the distribution of heat in such a manner as to effect formation heating that would enable maximum recovery by steam injection. Typically this would entail uniform heating of large oil sand volumes with the bulk of the heat concentrated in the layer with the highest content of oil. The heat distribution is strongly dependent on the actual current distribution in the formation since the thermal conductivity of oil sand is too low to allow for substantial heat flow. Several techniques have been proposed to facilitate favorable current distribution. One such method calls for the placement of vertical electrodes alternately above and below the richest layer of oil sand in order to take advantage of the higher electrical conductivity that is typically found in these adjacent strata (Chute *et al.*, 1987). The higher conductivity tends to spread out the current through these strata before it passes through the layer of rich oil sand, thereby heating the latter relatively uniformly. Other methods include the use of various configurations of electrode arrays which attempt to direct current paths in the desired manner throughout the formation (Chute, Vermeulen, McPherson, Hiebert & Fearn, 1985).

Although these techniques can contribute towards more effective preheating of oil sand formations, their potential has not been fully exploited because they have relied on current paths generated by single-phase excitation. With single-phase excitation, a.c. voltages of equal magnitude but opposite polarity (180 degrees out of phase) are used to excite the electrodes. Thus, for an array of electrodes excited with a single-phase source, only two distinct voltages are involved. Current will only flow between pairs of electrodes in the array that exhibit a difference in potential. Electrodes that are of the same potential will generate no current flow through the formation between them. Formation regions through which there is no current flow must rely on the very slow process of thermal conduction for heating. Typical compensation involves the close spacing of those electrodes that are excited at the same potential which again contributes to higher emplacement costs.

Areas of little or no heating and the use of close electrode spacing could be avoided and perhaps more uniform heating achieved if current pathways could be established between all electrodes in a given array. Such a current distribution can be generated by exciting the electrodes with voltage of equal magnitude but unequal phase (Pritchett, 1976). The magnitude of the potential difference established between each electrode pair will be determined by their phase relationship. For two electrodes excited with the same magnitude of voltage, but differing in phase by 180 degrees (like single-phase), the magnitude of the corresponding complex potential difference would be a maximum. Increasing or decreasing the phase separation from 180 degrees would reduce the magnitude of the potential difference and the current flow between the electrodes correspondingly.

The idea of using low frequency multiphase excitation to heat underground formations has appeared in patents in Canada and the United States. As part of his claims to a complete subterranean heating system, Yasuda (1978) outlined a type of switched, three-phase excitation design. His system consisted of six electrodes in the form of a hexagonal array with one central electrode (seven-spot pattern). Three-phase, delta and wye connections were then used to alternately excite different combinations of three and four electrodes within the array. In this manner, four unique patterns of current flow within the formation could be generated. By switching between the different circuits, distribution of electric current is spread throughout the volume to heated. Such a system, Yasuda proposed, would generate uniform heating. Pritchett (1976) presented similar ideas in his patented description of a switched single-phase electrode heating arrangement. He also suggested that similar heat distribution could be provided more efficiently by use of an array of multiphased electrodes that are continuously excited. As an example, Pritchett described current flow between electrodes in a formation as a function of time for a four-phase design. In doing so, he presented some evidence that a more uniform distribution of current would be provided with four-phase excitation than with single-phase excitation of the same configuration. He also illustrated current flow pathways for three-phase and eight-phase excitation.

Published works describing actual investigation and testing of the ideas presented in the aforementioned patents appear limited to date. The PCEJ group conducted a pilot project from April 1981 to April 1982 in the Athabasca oil sands that consisted of four vertically drilled electrode/producer wells defining an area 30 meters square (Khosla & Towson, 1985). The power source was a 2500 kVA three-phase supply. The four wells were initially connected so that electrodes one and four were jointly excited by one phase and electrodes two and three were excited by the other two phases. After approximately 75 days, the project was converted to a three well, three-phase operation for the duration of the project. No analysis of the multiphase excitation with regard to the heating patterns generated appears in the open literature.

Present Investigation

It is the goal, then, of this investigation to numerically examine some of the types of heating patterns that can be generated by low frequency, multiphase excitation. Several different electrode array configurations are studied. Among these are the four-spot, five-spot and seven-spot patterns which are well placement patterns commonly used in the oil industry. These array patterns can be repeated to enable heating of large volumes of oil sand. For these configurations, multiphase excitation with three phases is emphasized since it is readily available and its use would not require phase conversion equipment. In addition to the four-spot, five-spot and seven-spot arrays, circular arrays of vertical electrodes are also studied. These arrays are intended for use in heating a volume which is small enough to be confined within the array itself since the array pattern can not be repeated to form a larger array. Phasing schemes for the circular arrays are such that each electrode is at a unique phase angle with opposing electrodes in the array 180° out of phase. Heating patterns produced by circular arrays involving different numbers of electrodes with a corresponding number of phases are examined.

While emphasis is placed on the heating of oil sand *in-situ*, heating of underground, heavy oil formations around a single well is also investigated. Heating of underground formations via a single well is typically used in situations where oil flow is possible but impeded either due to a higher viscosity of the oil or due to wax buildup (Spencer, Bennett & Bridges, 1988). By heating the formation near the well bore and thereby increasing the flow, the pressure distribution in the formation is altered so that production from these wells is increased substantially (Bridges, Sresty, Spencer, & Wattenbarger, 1985; Gill, 1983).

Single well heating schemes that have been proposed in the past are similar in principle to multiple well heating schemes. The difference is that instead of current flow between the electrodes of different wells, the circuit was completed by flow of current from an electrode at the end of a single well back through the earth to a ground rod or the upper casing of the well (Stroemich, Vermeulen, Chute & Sumbar, 1990). With only a single electrode providing the electrical energy, the volume of heated formation is relatively small.

The single well scheme that delivers power to an underground formation most efficiently is the one where current returns to the upper casing of the well (Stroemich *et al*, 1990). This is referred to as the top/bottom configuration. With the top/bottom configuration, heating tends to be nonuniform along the length of the electrode since much higher current densities occur at the top end of the electrode (nearest the upper casing) than elsewhere along the electrode. This is especially true when heating thicker formations where longer electrode lengths are required.

Multiphase excitation can be applied to a single well if the single electrode is divided into three or more smaller electrodes, each separated by an electrically insulated section of well bore. Excitation of each electrode section by voltages with unequal phase angles would stimulate current flow between all electrode sections along their combined length. In this manner, the majority of the current would be contained in a volume near the electrode. Since current is responsible for heating, it is desirable to restrict current flow to the zone to be heated in order to make efficient use of the electrical energy. As well as increasing the containment of heating, it may also be possible to alter the phase separation between electrodes for the purpose of achieving greater heating uniformity or to focus heating in a particular zone.

Theory

The analysis of the advantages and limitations of multiphase excitation was carried out using computer simulation. The simulator program, EPEIOS, by A.D. Hiebert (1986) was designed to model low frequency electromagnetic heating of oil sand and other lossy material by numerically solving Maxwell's equations and the heat equation.

Before Maxwell's equations are solved, they are simplified. Simplification involves making approximations based on the nature of the fields created by low frequency excitation of electrodes embedded in oil sand. In their general phasor form, Maxwell's equations appear as:

$$\nabla \times \mathbf{E} = -j\omega \mathbf{B} \qquad (1.1)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + j\omega\mathbf{D} \qquad (1.2)$$

where $\mathbf{E}$ is the electric field intensity [$V \cdot m^{-1}$], ω is the angular frequency [$rad \cdot s^{-1}$], $\mathbf{B}$ is the magnetic flux density [T], $\mathbf{H}$ is the magnetic field intensity [$A \cdot m^{-1}$], $\mathbf{J}$ is the electric current density [$A \cdot m^{-2}$] and $\mathbf{D}$ is the electric flux density [$C \cdot m^{-2}$]. At low frequencies however, the magnetic fields created by the time varying electric fields in the formation may be neglected. Thus, equation (1.1) may be rewritten as:

$$\nabla \times \mathbf{E} \cong 0 \qquad (1.3)$$

from which it follows that

$$\mathbf{E} = -\nabla\mathbf{\Phi} \qquad (1.4)$$

where $\mathbf{\Phi}$ is the electric scalar potential. This approximation, known as the quasi-static approximation, is valid provided that the wavelength is much greater than the system dimensions. For the case of oil sand heating at 60 Hz, the wavelength in the formation is 2000 m to 9000 m, depending on moisture content (Vermeulen *et al.*, 1979). The largest system dimension will typically be less than 200 m.

Further simplification is possible when one considers the loss tangent. It is defined as $\frac{\sigma}{\omega\varepsilon}$, where σ is the electrical conductivity [$S \cdot m^{-1}$], and ε is the electrical permittivity [$F \cdot m^{-1}$]. The loss tangent represents the ratio of conduction current to displacement current in the system. In oil sand at 60 Hz the loss tangent is much greater than unity. This indicates that the conduction current, $\mathbf{J} = \sigma \mathbf{E}$, dominates and the displacement current may be neglected. Equation (1.2) may then be rewritten as:

$$\nabla \times \mathbf{H} \cong \mathbf{J} \qquad (1.5)$$

from which it also follows that:

$$\nabla \cdot \mathbf{J} \cong 0 \qquad (1.6)$$

Equations (1.3), (1.4), (1.5) and (1.6) are the defining equations for the electromagnetic fields and currents created by low frequency excitation in oil sand. Use of Maxwell's equations in this form greatly reduces their complexity, thus making them easier to solve. However, it does limit the scope of the simulator to cases which support the simplifying assumptions i.e. cases where the quasi-static approximation is valid and the loss tangent is much greater than one.

The solution to the electrical equation is obtained by combining equations (1.4) and (1.6) to yield a form of the current continuity equation in terms of electric scalar potential.

$$\nabla \cdot (\sigma \nabla \Phi) = 0 \qquad (1.7)$$

This equation is then solved for specified boundary conditions in three dimensions using a seven point finite difference approximation method. In using the finite difference technique, the partial differential equation is transformed into a system of algebraic equations which can be easily solved.

Considering the case of multiphase excitation, the electric scalar potential becomes a complex value possessing both real and imaginary components. A solution is obtained in this instance by separating equation (1.7) into two equations, one for the real potential and the other for imaginary potential. This is possible since the vector operators are linear and the electrical conductivity is always real (Hiebert, 1986). The method of solution for the two equations is identical except that the boundary conditions at the electrodes will be different. When solving for real potential, the rms values of the real components of complex excitation voltage are assigned to the electrodes. Similarly, when solving for imaginary potential, the imaginary components of the excitation voltages are used at the electrodes.

The average heating rate is determined from the electric field using the equation:

$$Q = \sigma \, \mathbf{E} \cdot \mathbf{E}^* \qquad (1.8)$$

where Q is the heating rate [$\mathrm{W \cdot m^{-3}}$]. Assuming no fluid flow and heat transfer via conduction only, the temperature distribution can be calculated using:

$$M \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + Q \qquad (1.9)$$

where M is the volumetric heat capacity of the material [$\mathrm{J \cdot m^{-3} \cdot K^{-1}}$], T is the temperature [°C] and k is the thermal conductivity [$\mathrm{W \cdot m^{-1} \cdot K^{-1}}$]. This equation is also solved using a seven point finite difference formulation.

Electrical and thermal equations are non-linearly coupled by the temperature dependence of the electrical conductivity of oil sand. This dependence is represented by the following equation:

$$\sigma = C_{24} + \alpha_b (T-24) + \alpha_c (T-24)^2 + \alpha_d (T-24)^3 \quad (1.10)$$

where C_{24} is the electrical conductivity [S] of the modelled medium at 24°C , and α_b , α_c, and α_d are experimentally determined coefficients (Stroemich, Sumbar, Vermeulen, & Chute, in press). The coupling of the electrical and thermal equations is accounted for by sequentially solving each set of equations for discrete time steps. Time step sizes are automatically calculated by the program based on a specified, limited change of temperature and electrical conductivity during the time interval.

Results produced by the simulator have been extensively verified by comparison to analytically solved problems and to results obtained from physical scale models (Hiebert, 1986). While the basis simulator used in the present investigation is the one developed by Hiebert, the author made considerable changes to the code and adapted it to a Macintosh environment. Appendices A, B and C detail these modifications.

CHAPTER 2

Details of the Investigation

As mentioned previously, computer simulation was used to analyze the types of heating patterns that could be generated by low frequency multiphase excitation of electrodes embedded in oil sand or similar media. The simulation program EPEIOS was initially developed for use on the University of Alberta mainframe computer (Amdahl) which ran under the MTS operating system.

Although the Amdahl was able to provide the computing power necessary to perform complicated EPEIOS simulations quickly, the MTS operating system was inconvenient to work with. The use of CPU time was restricted and graphical capabilities were limited. In addition, the user had to constantly monitor the system for changes and updates that could affect program execution.

It was felt that the inconveniences of the MTS system could be avoided if the EPEIOS program was transferred to a microcomputer. The recent rapid advance of microcomputer technology has produced machines that are both powerful and extremely flexible given the support of available software. Unfortunately, the microcomputer systems that were available to serve as host to the EPEIOS program possessed computing speed that was somewhat less than that of the Amdahl. However, it was anticipated that the computational capabilities of microcomputers would continue to increase and that faster machines would soon be developed. Once the EPEIOS program was established on one microcomputer, it would be very easy to move it to a more powerful model when it became available. Thus, after consideration of the intended use of the EPEIOS simulation program, both for this study and for future studies, the decision was made to first transfer EPEIOS from the Amdahl to a microcomputer system before beginning multiphase simulations.

The microcomputer to which the EPEIOS program was transferred was an Apple Macintosh SE/30. It was chosen because it could provide an acceptable level of computing power and it was able to quickly and easily produce desired graphical output using available software. In addition, the Macintosh possessed an advanced user interface system which made it extremely easy to use.

In transferring the EPEIOS program from the Amdahl to the Macintosh SE/30, several modifications were made to the program code. Differences in compilers between the two systems had to be accounted for and data files compatible with Macintosh graphics programs had to be developed. These changes are documented in detail in Appendix A.

After the transfer process was completed, further modifications were made to the program on an ongoing basis. These changes were initiated by difficulties and inconveniences that were encountered as the EPEIOS simulator was used. New control variables were added, the interpolation subroutine was improved and grid structure limitations were expanded. In addition, a new version of the program was created that was capable of being compiled and executed on a high speed coprocessor that had been installed in an available Macintosh II. Use of this version reduced simulation run times by 15 - 20 times. For completeness, all developmental modifications are documented in Appendix B.

EPEIOS simulates low frequency electromagnetic heating by solving the governing electrical and thermal equations throughout the volume of the modelled configuration. As mentioned in the theory, these equations are solved by a finite difference approximation technique. The seven point method used in EPEIOS was designed for three dimensional problem domains that are represented as a composition of grid blocks. Although the sizes of the blocks may be varied, all grid blocks are rectangular prisms with sides parallel to the coordinate axes.

In designing a configuration for simulation, it is important to choose the grid structure carefully. A fine grid structure made up of very small grid blocks will yield a more accurate solution than a coarse grid structure composed of large grid blocks. This is because the smaller blocks provide a better finite difference approximation to the actual partial differential equations. However, the use of small grid blocks requires that more blocks be used in the problem domain. This increases the number of algebraic equations that must be solved to obtain a solution. As a result, the computational time required to perform a simulation increases as well. The fewer the number of grid blocks that are used, the less time it will take to perform the simulation. In an effort to keep simulation run times at a reasonable length while maintaining the accuracy of the results, grid structures composed of varying sizes of grid blocks were designed for each configuration after consideration of the nature of the fields that would be created. In regions where there was rapid variation in electric potential (such as near electrodes), the volume was divided into relatively small grid blocks. This was necessary because a second order polynomial is used by EPEIOS to obtain an approximation for electric field strength at the interface between two grid blocks. This approximation would be inaccurate in a rapidly varying field if the centers of the grid blocks were far apart. Conversely, in regions where there was little variation in electric potential, larger grid blocks were used.

Several configurations that were studied were completely uniform in one direction. In those cases, the problem became two dimensional and only two grid blocks were needed in that direction.

Although the ability to vary the dimensions of grid blocks gave the simulator flexibility, the rectangular shape of the blocks necessitated the approximation of nonrectangular geometries.. Electrodes which in reality would have a circular cross section were modelled as having a square cross section. The length of one side of the square electrode was made to equal the diameter of the modelled electrode. This appeared to give a good approximation provided that system dimensions were much larger than the electrode diameter.

It should also be mentioned that with the finite difference scheme employed by EPEIOS, results can be sensitive to the chosen orientation of the grid structure with respect to lines of current flow within the problem domain. This is due to the nature of the seven point finite difference approximation that is used to approximate the current continuity equation. With the seven point technique, difference equations are built around each block by considering the six surrounding blocks which share a common face. Three points are used in each direction (x,y,z) to create finite difference equations which approximate the differentials in the x, y and z directions. In this manner, current flow through any one block is considered to be along the coordinate axis. Diagonal flow of current through a block is not effectively simulated since corner blocks are not included in the difference equation. As a result, current flow that follows a path diagonal to the grid structure could yield different heating rates than those produced by the same current flowing along one of the coordinate axes. This effect, known as a grid orientation effect, has also been encountered and examined by researchers using the finite difference technique to model reservoir fluid flow (Settari & Karcher, 1985).

In studying multiphase excitation of electrode arrays, paths of current flow diagonal to the orientation of the grid structure could not be avoided. Consequently, grid orientation effects were a concern. One possible solution would be to replace the seven point finite difference equations with higher order equations which included a greater number of points around each grid block. However, doing this would have been a very complex and time consuming task. In addition, such a change would greatly increase the number of computations that would have to be performed for a given simulation. This would result in increased run times. Further investigation into grid orientation revealed that the effects could be reduced by dividing the problem domain into very small grid blocks. Once again, in doing this, a better approximation of the exact differential is obtained. In light of the investigative nature of this study, it was felt that sufficient accuracy could be achieved in this manner.

Another difficulty related to grid orientation was discovered when attempting to model a nonrectangular array of electrodes as a subarray of a much larger array. When modelling this type of problem, no-flow boundaries correspond to vertical planes on the sides of the subarray. Thus, for a nonrectangular subarray, one or more of the no-flow boundaries will not coincide with one of the three major planes (x-y, y-z and x-z). No-flow boundaries that do not coincide with one of the major planes have to be approximated as a path defined along the edges of grid blocks as shown in Figure 2.1.

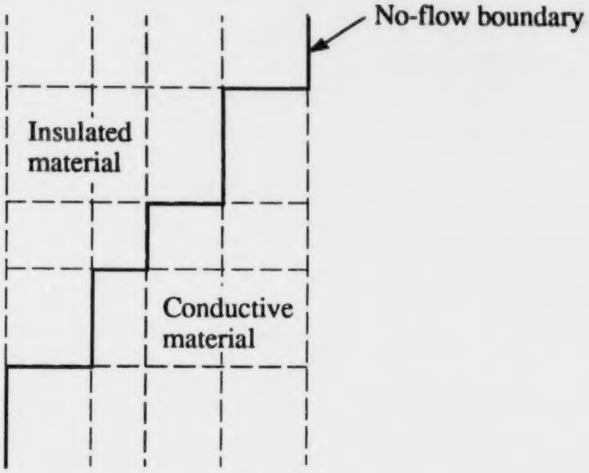


Figure 2.1 Approximation of No-Flow Boundary Diagonal to Grid Orientation.

In performing a simulation involving diagonal boundaries, results showed obvious distortion of heating patterns along the jagged no-flow boundaries when compared to heating patterns along the no-flow boundaries coinciding with one of the major planes. The distortion of the heating patterns suggested that current flow along diagonal boundaries was not properly simulated. Reduction of the size of grid blocks that formed the boundaries improved the results. However, inaccuracies were still apparent when the maximum number of grid blocks allowed by the program were used in the simulation. When the simulation was repeated with rectangular no-flow boundaries seemingly placed at infinity, heating along diagonal paths was more comparable to heating along paths parallel to the grid orientation. The conclusion was reached that the approximated diagonal boundaries of no-flow further restricted the ability of the simulator to model current flow that follows a path diagonal to the orientation of the grid structure. For this reason, array configurations which outlined nonrectangular shapes were not examined as being a subarray in a much larger array. Instead, these subarrays were modelled as being complete in themselves. No-flow boundaries were rectangular and were established far enough away from electrodes that they did not significantly affect current flow in the array.

To perform a simulation, the modelled configuration had to be described in terms of the input variables employed by the EPEIOS program. Values assigned to these variables are read into the program at the time of execution in namelist format from a single file. Descriptions of the variables as well as information on running EPEIOS on a Macintosh computer can be found in the user's manual. This manual is included in this work as Appendix C.

The focus of this research was to investigate applications of multiphase excitation to methods of ohmic heating of materials via embedded electrodes. To do this, multiphase excitation of an electrode configuration was compared to single-phase excitation of the same configuration with input power to the electrodes kept constant.

In general, the basis for comparison was chosen to be heating rates generated within the material. The reasons for this were twofold. First of all, in considering heating rates, results became independent of the thermal characteristics of the heated formation. Values for thermal conductivity and thermal capacity which are needed to calculate temperature, are not considered in the calculation of heating rates. Eliminating the dependence of the results on the thermal characteristics of the material allows for more generalized interpretation of the results. The second reason for considering heating rates was that values could be calculated in one time step. This helped minimize the time it took to perform a simulation. The exception was the case of the single well configuration where temperature distributions were used to clarify the heating rate patterns that were observed.

Further generalization was achieved by specifying the heated formation as homogenous and isotropic for most simulations. The exception is once again the single well configuration where a formation consisting of two layers with different electrical conductivities was considered. Heating rates are a product of electrical conductivity and the square of the magnitude of the electric field intensity (refer to Equation (1.8) in Chapter 1). For low frequency excitation in a conductive isotropic material, the generated electric field is not influenced by the characteristics of the material. In addition, electrical conductivity becomes a scalar quantity for isotropic media. Thus initially, the heat distribution within a formation is not affected by its electrical conductivity. Of course the value specified for conductivity will directly influence the magnitude of the heating rates throughout the formation. Also, as heating progresses, the temperature dependence of the electrical conductivity will cause the formation to become anisotropic as a result of the different heating rates that are generated. However, this is not a concern since the comparative nature of this study places the emphasis on the initial relative magnitudes of the heating rates for differing excitation schemes.

For most simulations the heated material was chosen to be Athabasca oil sand that possessed a relatively high content of oil (rich oil sand). It was modelled as being electrically and thermally conducting with an initial temperature of 10°C. Parameters that were used to characterize the material were thermal conductivity and thermal capacity and electrical conductivity as a function of temperature. Although thermal characteristics are not required in the calculation of heating rates, it is necessary that these parameters be specified for all regions in each simulation in order for the simulator to run properly. Values for these parameters were obtained from testing previously done at the Applied Electromagnetics Laboratory at the University of Alberta (Stroemich *et al.*, in press).

Electrodes were modelled as perfect conductors. Values for thermal conductivity and thermal capacity, which were again required for all simulations, were set equal to those of copper. Electrode dimensions were arbitrarily chosen based on typical values that would be used in a practical implementation scheme. In general, electrode spacings were decided upon as a matter of convenience with consideration of grid structure limitations (the number of grid blocks and grid orientation).

When simulating configurations involving several electrodes in some form of array, electrode lengths were set equal to the thickness of the heated formation. By not considering a formation thickness greater than the electrode lengths, the problem domain became two dimensional since it was completely uniform in the direction parallel to the length of the electrode. The two dimensional nature of the problem made for easier solution of the governing equations. This also helped to keep simulation run times down since fewer iterations were required to arrive at a solution. In addition, the uniformity of the problem domain in one dimension enabled the use of a minimum number of grid blocks (two) in that direction. As mentioned previously, minimizing the number of grid blocks that are used to model a configuration helps minimize the time it takes to perform the simulation.

Designing array configurations to be two dimensional problems did not adversely affect the analysis of multiphase excitation. This is because differences in excitation schemes would not significantly alter heating patterns along the vertical plane. Changing the excitation of the electrodes from single-phase to multiphase would only serve to alter the paths of current flow between the electrodes. Thus, it was only current flow paths (heating rates) in the transverse plane that were of interest.

It was felt that the patterns of heat distribution for the various configurations could be best represented by greyscale plots. From data files generated by EPEIOS, the plots were created using a commercially available graphing package called Spyglass™. It should be pointed out that in producing the plots, some interpolation between data points was carried out by the graphing program in order to generate an image. Although such interpolation would not substantially alter results, it would impose somewhat of a smoothing effect in places of rapid variation in magnitude of the plotted quantity.

For a typical group of data points, the range in values of heating rate is very large. Heating rates near the electrodes are much greater (two orders of magnitude) than those found throughout the rest of the formation. If for a greyscale plot, the scale included maximum and minimum values of heating rate, the grey scale would be very coarse since there is a limited number of discrete shades of grey. Use of such a coarse scale would have resulted in graphs that provided little comparative information. A coarse scale would not be able to display the relatively fine differences in heat distribution which are present in the bulk of the formation. It was these fine differences that most clearly showed the effects of changes in electrode excitation. Consequently a narrower range of values was used for the greyscales. The chosen range was representative of heating rates found within the formation at some distance from the electrodes. All data values that exceeded the range of the greyscale are then represented by the greyscale corresponding to the maximum value in the range. Thus the existence of relatively high heating rates near the electrodes is not directly conveyed by the greyscale plots.

CHAPTER 3

Results of the Computer Simulations

The results of the computer simulations are presented here in order of electrode configurations. Arrays used for the purpose of heating large volumes of oil sand *in-situ* are examined first. These results are followed by an examination of circular arrays which may be used for heating smaller volumes of formation. Circular array configurations are quite similar to the arrays proposed for heating large oil sand formations. The electrode arrangements used for the single well are somewhat different from all other configurations studied in this thesis. Consequently, the results of simulations involving the single well are presented last.

The Four-Spot Array

The four-spot configuration is a standard vertical electrode placement pattern that consists of four electrodes in a square array. It can be repeated to encompass large areas of formation as shown in Figure 3.1. When used in this capacity, planes of symmetry that are parallel to the electrodes may be defined along each row and column of electrodes. These planes of symmetry indicate planes, on either side of which the problem domain is electrically and thermally the same. The planes of symmetry effectively divide the large array consisting of the repeated four-spot pattern into a square grid with the electrodes marking the corners of each square. The solution within each volume bounded by the planes of symmetry would be the same. No-flow boundaries that are electrically and thermally insulating may be used to represent the planes of symmetry since there will be no current flow or heat flow across them. Thus, heating generated by a large array consisting of the repeated four-spot pattern can be determined by finding the heating within the four-spot subarray surrounded by no-flow boundaries. A sample region within the large array is indicated in Figure 3.1.

A plan view of the electrode placement and the corresponding no-flow boundaries that are used in the simulation is shown in Figure 3.2(a). Notice that the no-flow boundaries intersect at the center of each electrode. Since the problem domain was defined within these boundaries, electrode volumes used in the simulation represented only one quarter of the actual electrode volumes.

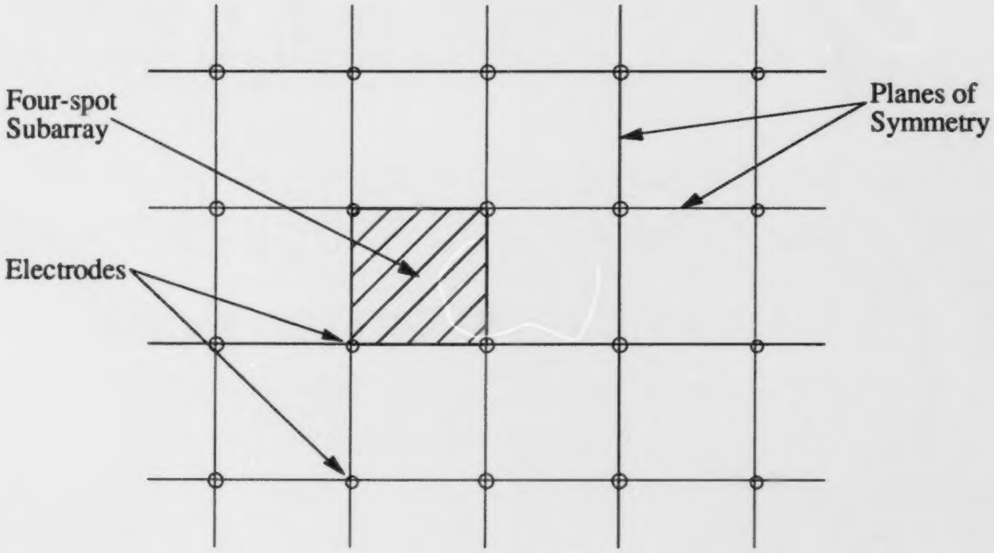


Figure 3.1 Large Array Formed From Repeated Four-Spot.

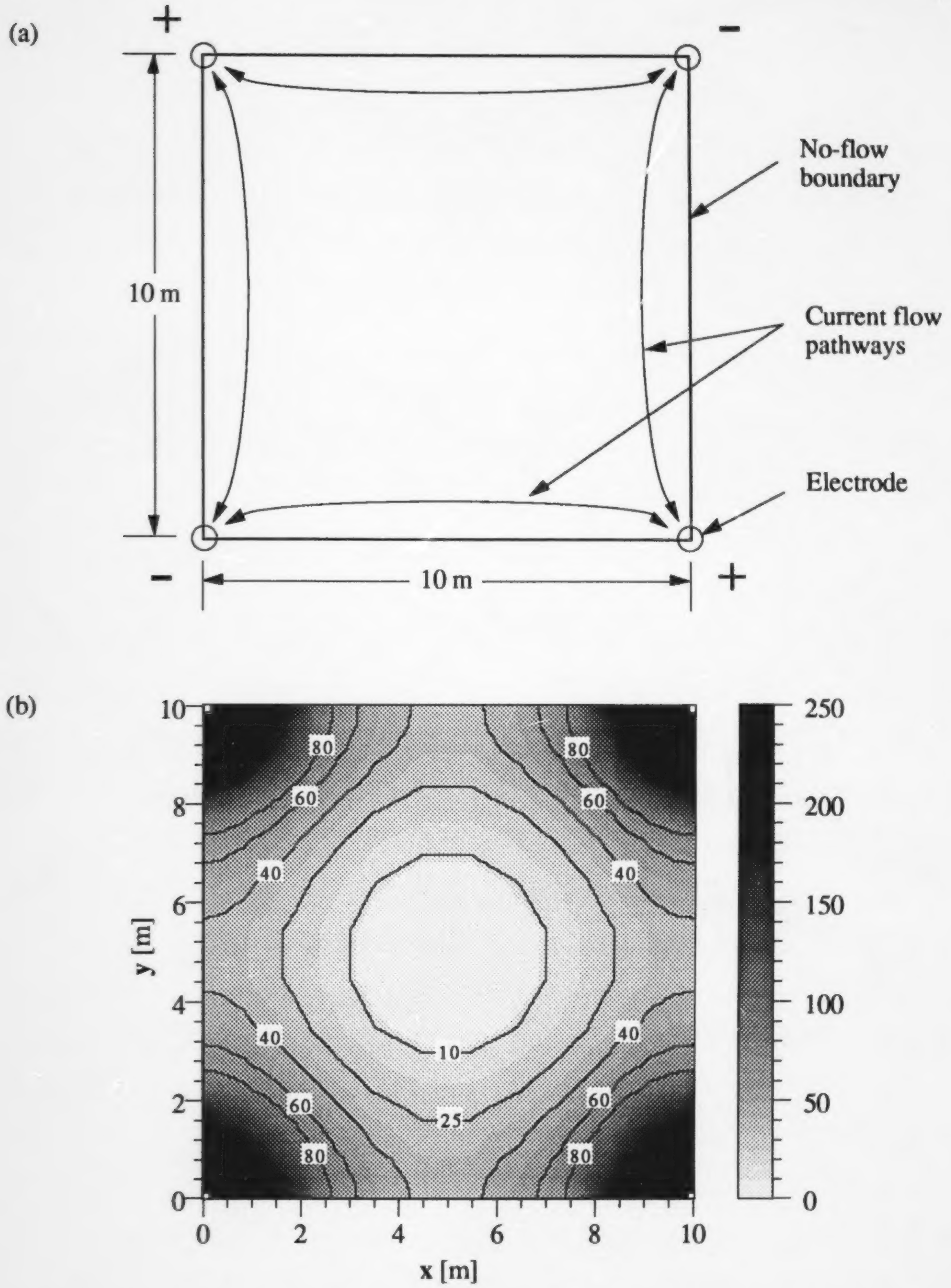


Figure 3.2 (a) Plan View of Modelled Four-Spot Subarray with Polarities and Current Pathways Shown for the First Single-Phase Excitation Scheme. (b) Heating Rates [$W \cdot m^{-3}$] in Horizontal (x-y) Plane for Four-Spot Subarray with Single-Phase Excitation Shown in (a). Total Input Power Is 50 kW.

A grid block representation of the four-spot configuration was constructed. Coordinate axes were chosen such that the vertical electrodes were parallel to the z direction. Electrode spacing along the sides of the square array was chosen to be 10 meters. Formation thickness as well as electrode lengths were specified as equal to 5 meters so that the problem would be two dimensional (uniform in the z direction). The radius of each electrode was 0.2 meters. The formation material was specified as having an electrical conductivity equal to $1.1\text{x}10^{-3}$ $\text{S}\cdot\text{m}^{-1}$.

Using EPEIOS, single-phase and four-phase excitation of the described four-spot configuration was simulated. To better emphasize the contrast in current distribution, two different single-phase schemes were compared to the multiphase scheme. For all excitation schemes, total input power was a constant 50 kW. A comparative analysis is as follows.

For the first method of single-phase excitation of the square array, two electrodes were specified as having positive polarity (phase angle of 0°) and two were specified as having a negative polarity (phase angle of 180° with respect to the positive electrodes). Electrode potentials were assigned such that adjacent electrodes around the array had opposite polarities. As shown in Figure 3.2(a), such an assignment meant that the potential of the electrodes alternated between positive and negative polarities along a path that followed the perimeter of the problem domain.

Although polarity differed, the relative magnitudes of the potential at each electrode were specified as being the same. The actual values of the magnitude were automatically adjusted by the EPEIOS program to ensure that the specified total input power was kept constant (in this case 50 kW).

Also shown in Figure 3.2(a) are the paths of current flow that would be generated by the single-phase electrode excitation described. Although they are drawn as single lines, current would actually fan out along many different paths between electrodes. As can be seen, these paths only appear along the perimeter of the array between electrode pairs with opposing polarities. The amount of current that flows along each path would be equal since the magnitudes of the electrode potentials are equal, the electrode separation distance is equal and the medium is isotropic. There would be no current flow between diagonally opposing electrodes since they are at the same potential.

A greyscale plot of heating rates within the formation in the x-y plane is shown in Figure 3.2(b). Due to the uniformity of the problem domain in the z direction, this data slice would be the same for any z coordinate. As was expected, given the relationship between low frequency current and heating rate, the distribution of heating rate is a reflection of the distribution of the square of the current magnitude.

The highest heating rates were observed in regions near the electrodes. Here, conduction current density (which is directly proportional to the square root of the heating rate) would have been the highest since all current flow in the formation originates from these points. The decrease in heating rate with the radial distance from the electrode is a result of reduced current density as current spreads out into the formation.

Away from the electrodes, the only other regions of significant heating were strips between the electrodes along the no-flow boundaries. Heating in these regions corresponds to the paths of current flow along the boundaries as shown in Figure 3.2(a).

In the central region of the array there appeared to be little or no heating. The lack of heating in this region can be attributed to the absence of current flow between diagonally opposing electrodes. Without diagonal current flow, the conduction current densities were relatively low towards the center of the array and consequently little heating occurred in this region. With time, an increase in formation temperature at the center of the array would take place due to the effects of thermal conduction. However, when considering oil sand as the formation material, thermal conduction is a relatively slow process. This means that the rate of temperature increase at the center of the formation would be considerably less than in regions near the electrodes where heating rates were relatively high. If this form of single-phase excitation were used as a means to raise the temperature of a large formation substantially, overheating at the electrodes would be a major concern. This would be especially true if greater electrode spacing were used.

The second single-phase electrode excitation scheme also consisted of two positive and two negative electrodes. For this arrangement however, electrode polarities were assigned such that a potential difference was established between diagonally opposing electrodes. As a result of the change in electrode excitation, the paths of current flow within the formation were different from those for the previous single-phase scheme. The opposite polarities between diagonally opposing electrodes caused diagonal current flow. In addition, current flowed between electrodes along two opposing sides of the square array. Since the same number of positive and negative electrodes were used for this scheme as for the first one, the number of current flow paths were also the same. The polarity assignment and the resulting paths of current flow are shown in Figure 3.3(a).

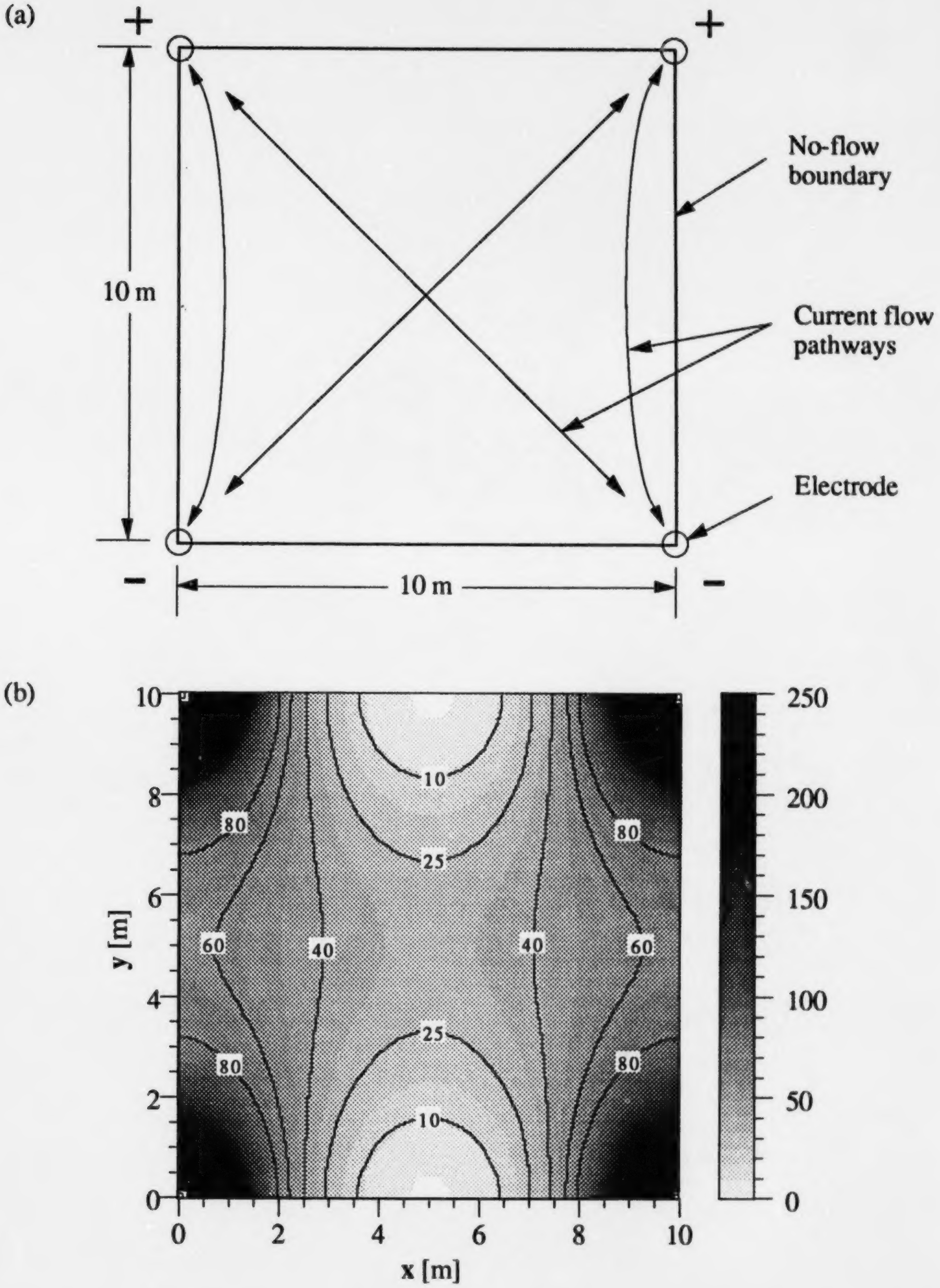


Figure 3.3 (a) Plan View of Modelled Four-Spot Subarray with Polarities and Current Pathways Shown for the Second Single-Phase Excitation Scheme. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Four-Spot Subarray with Single-Phase Excitation Shown in (a). Total Input Power Is 50 kW.

A greyscale plot of the heating rates in the x-y plane for an arbitrary z coordinate is shown in Figure 3.3(b). It is readily apparent from the plot that the change in electrode polarities significantly altered the distribution of heat.

As in the previous single-phase scheme, the highest heating rates occurred in regions close to each of the four electrodes. Also as in the first excitation scheme, there were regions of the formation for which there was little heating. As seen in the plot, these unheated regions were located along two sides of the array between electrodes with the same polarity. As before, these regions corresponded to regions through which there was no direct path of current flow.

Unlike the first single-phase excitation scheme the intensity of the heating along each of the four current paths was not the same for this scheme. Much higher heating rates (approximately twice) were observed between adjacent electrodes on the two sides of the array than between diagonally opposing electrodes. This can be explained as follows. With single-phase excitation, the potential difference between all electrode pairs with opposing polarity is the same. In the case of the single-phase excitation scheme examined here, the potential difference between electrodes stimulating diagonal current flow would be equal to the potential difference between electrodes stimulating current flow along the two sides of the array. Since the distance between diagonally opposing electrodes is greater than the distance between adjacent electrodes on one side of the array, a greater resistance to current flow would exist along the diagonal path. Consequently, for equal potential difference, less current would flow diagonally through the formation than along either of the two sides which exhibit current flow. The lower magnitude of diagonal conduction current would mean lower current densities along the diagonal paths. This would result in the lower heating rates observed at the center of the array than along the two sides.

For multiphase excitation of the four-spot square array, all electrodes were once again assigned an equal magnitude of potential. However, each electrode was also assigned a unique phase angle. Excitation phase angles of 0°, 90°, 180° and 270° were used. Relative phase angles were assigned such that adjacent electrodes exhibited a phase separation of 90°. In this manner, diagonally opposing electrodes had a phase difference of 180°. Electrode potentials for the four-spot array are shown in Figure 3.4(a).

Although the magnitudes of the voltages at all electrodes were the same, the phase separation between electrodes created a complex potential difference which stimulated current flow. The magnitude of the potentials between a specific pair of electrodes was dependent on their phase separation. In the case of the four-phase excitation scheme studied here, a greater potential difference existed between diagonally opposing electrodes which had a phase separation of 180° than between adjacent electrodes which had a phase separation of 90°.

Four-phase excitation generated a greater number of current paths through the formation than either of the two single-phase schemes (six as opposed to four). As can be seen in Figure 3.4(a), current flow was between electrodes along the four sides of the array as well as along the two diagonal paths. The use of the four phases increases the number of current paths because no two electrodes are at the same potential. The phase angle of each electrode is unique and this causes each electrode to be

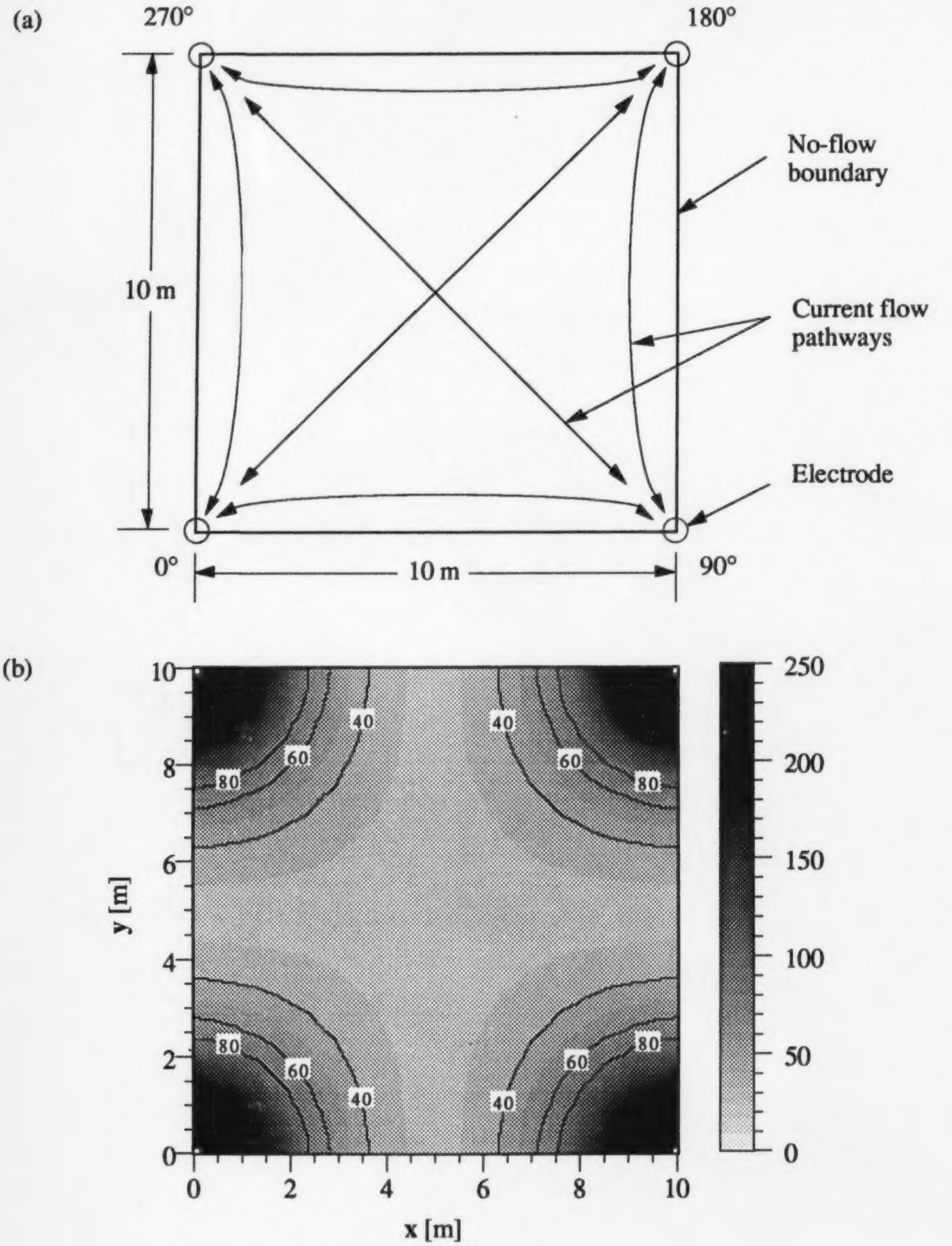


Figure 3.4 (a) Plan View of Modelled Four-Spot Subarray with Phase Angles and Current Pathways Shown for Four-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Four-Spot Subarray with Four-Phase Excitation. Total Input Power Is 50 kW.

in essence at a different potential than the other three.

Heating rates generated by the four-phase excitation in the x-y plane of the formation are shown in Figure 3.4(b). As with single-phase excitation, the highest heating rates were in the regions near the four electrodes. The symmetry of the heating pattern around each electrode would seem to indicate that current was evenly distributed on the entire surface of the electrode. The even distribution of current would help to minimize the high current densities at the electrodes which cause overheating.

In the bulk of the formation material away from the electrodes, heating was relatively uniform. Unlike single-phase excitation there were no regions where heating rates were significantly lower than in the rest of the formation. The uniformity of heating implies that current densities were uniform as well. This in turn suggests that similar magnitudes of current flowed along all current paths in the formation.

It is apparent that equivalent current flow would have existed in the four paths along each side of the array. This is because the electrode pairs on each side had excitation voltages that were separated by the same phase angle. Equal phase separation caused the magnitude of the potential differences which stimulate current flow to be equal as well. Thus for homogeneous and isotropic media, the magnitude of current flow along the four routes would be the same. It should however be noted that since four different phases of voltage excitation are used, current flow along the four outside paths would not all be in phase.

As mentioned previously, diagonal current paths are longer than those along the sides of the array. Thus the resistance to diagonal current flow is greater than the resistance to current flow between adjacent electrodes. If the potential difference between diagonally opposing electrodes was the same as the potential difference between adjacent electrodes, significantly less current would flow along the diagonal path as demonstrated by the second single-phase excitation scheme. With four-phase excitation however, the magnitude of the complex potential difference between diagonal electrode pairs is different from that between adjacent pairs. The greater potential difference between diagonally opposing electrodes compensates for the higher resistance. In this way, similar current flow is maintained between diagonal electrode pairs and adjacent electrode pairs. The result is uniform current distribution in the formation which generates the uniform heating seen in Figure 3.4(b).

The Five-Spot Array

The five-spot configuration is another well placement pattern that is often used in conventional oil recovery schemes. It is identical to the four-spot array except that a fifth vertical well is located at the center of the square region outlined by the other four.

Considered as an electrode placement pattern, the five-spot would typically be repeated numerous times to cover a large region of formation as shown in Figure 3.5. As discussed previously, the repetition of the pattern in this manner would create planes of symmetry around the array that can be modelled as electrically and thermally insulating boundaries. These no-flow boundaries which lie around the perimeter of the five-spot pattern define a rectangular problem domain as indicated in Figure 3.5. This allowed the five-spot configuration to also be studied as a subarray that can be repeated to form a much larger array.

For computer simulation of the five-spot configuration, a grid block representation of the electrode array buried in the formation was constructed. Coordinate axis were defined such that vertical electrodes ran parallel to the z direction. As before, the lengths of the electrodes were made equal to the formation thickness to keep the problem two dimensional. In the horizontal plane (x-y plane), adjacent corner electrodes in the five-spot pattern were separated by a distance of 10 meters. Due to the geometry of the array, the center electrode was then a distance of approximately 7.1 meters from each corner electrode. Electrodes were all modelled as having a radius of 0.2 meters. A plan view of the problem domain can be seen in Figure 3.6(a).

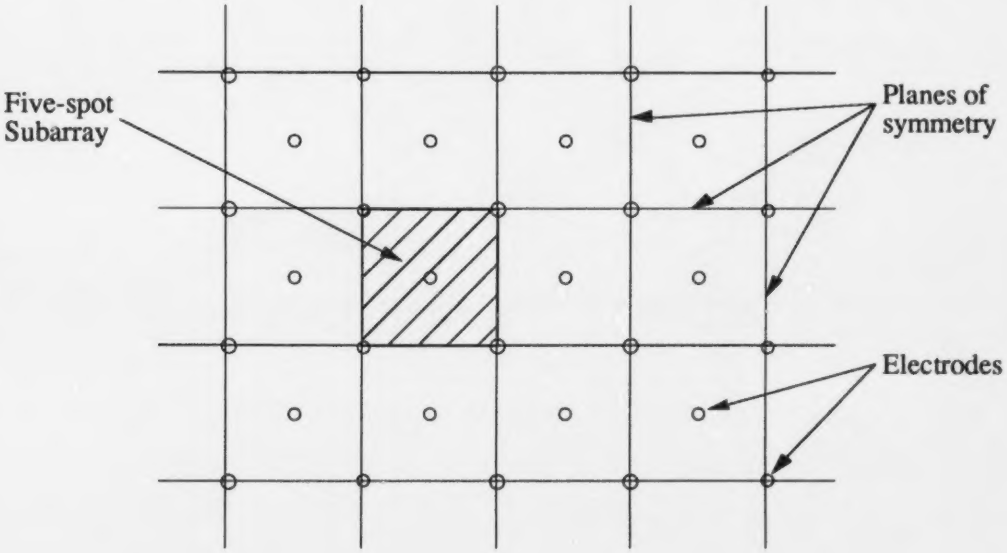


Figure 3.5 Large Array Formed From Repeated Five-Spot Pattern.

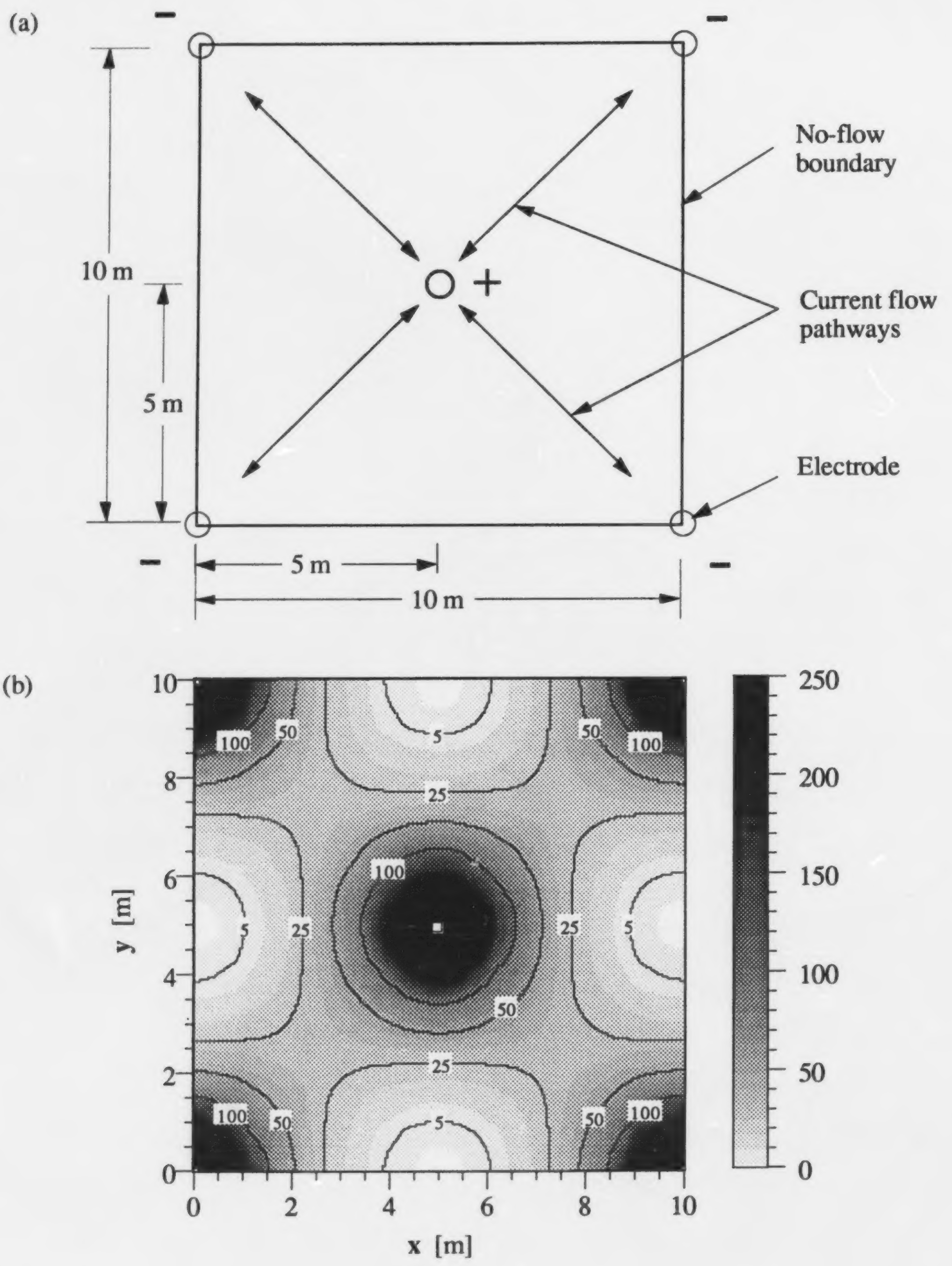


Figure 3.6 (a) Plan View of Modelled Five-Spot Subarray with Polarities and Current Pathways Shown for Single-Phase Excitation. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Five-Spot Subarray with Single-Phase Excitation. Total Input Power Is 50 kW.

With rich oil sand as the formation material (conductivity equal to $1.1\text{x}10^{-3}\ \text{S}\cdot\text{m}^{-1}$) and a constant input power of 50 kW, heating rates generated by single-phase excitation were compared to those generated by two different three-phase excitation schemes.

Single-phase excitation of the five-spot array consisted of equal magnitudes of potentials at all five electrodes. The polarity of the center electrode was positive and the polarity of the four corner electrodes was negative. This polarity assignment has previously been proposed as a single-phase excitation scheme for the five-spot array (Chute *et al.*, 1985). The various electrode polarities are shown in the plan view of the five-spot array in Figure 3.6(a).

Heat distribution within the formation along the x-y plane is shown in Figure 3.6(b). Heating away from the electrodes was limited to relatively narrow channels between the center electrodes and each of the surrounding electrodes. These regions correspond to current flow pathways between electrodes in the array. As would be expected, there was no heating along the sides of the problem domain between adjacent corner electrodes. These electrodes were at the same potential and thus they would not stimulate any current flow between them. Without current flow through the formation, there is no electric heating.

For the first three-phase excitation scheme, equal phase separation between all three phases was used. A phase angle of 0° was specified for the center electrode and phase angles of 120° and 240° were assigned to the other four such that adjacent corner electrodes were of differing phase. As shown in Figure 3.7(a), this phase assignment yields the equivalent of four delta connections with unbalanced loads. The loads are unbalanced

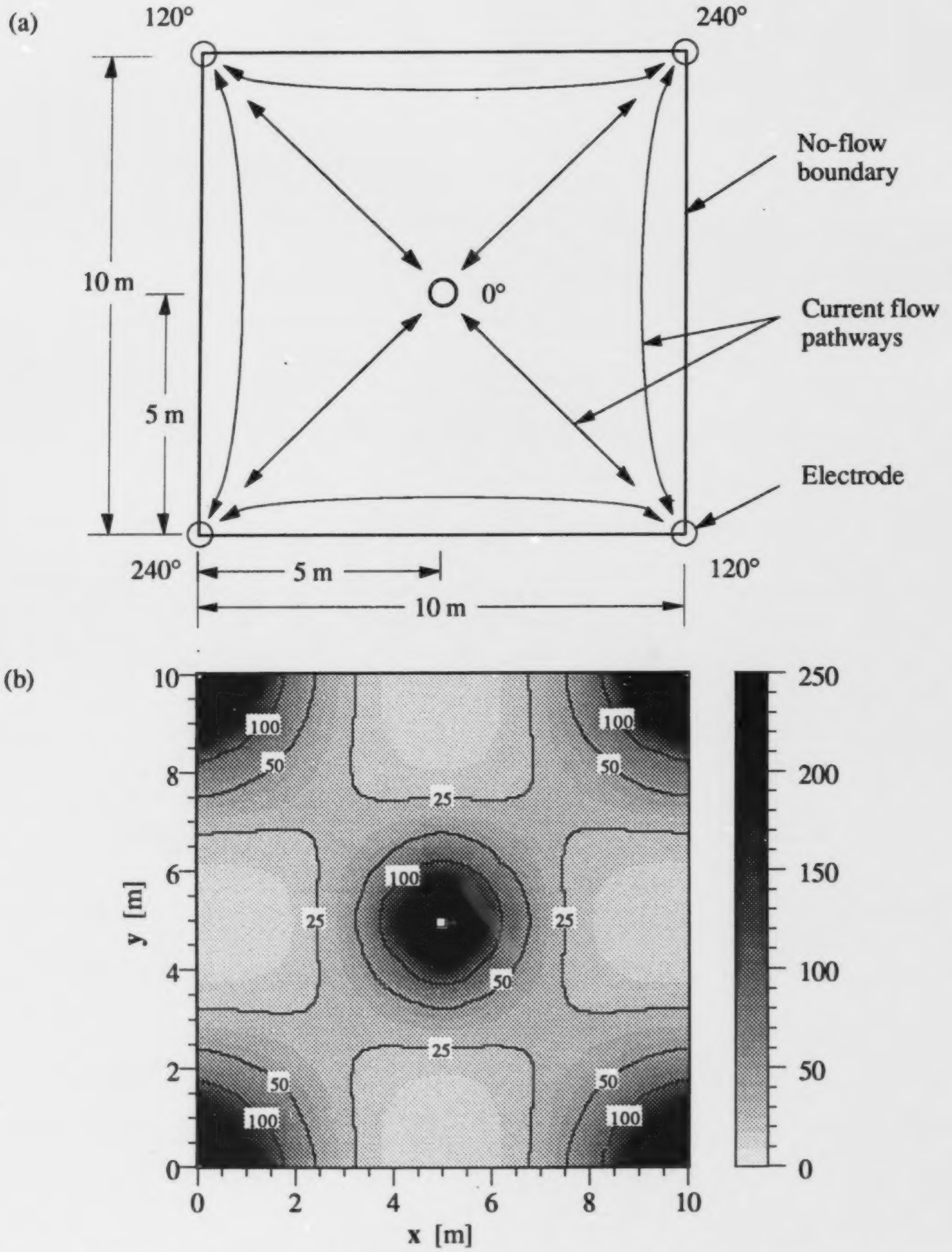


Figure 3.7 (a) Plan View of Modelled Five-Spot Subarray with Phase Angles and Current Pathways Shown for Standard Three-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Five-Spot Subarray with Three-Phase Excitation as Shown in (a). Total Input Power Is 50 kW.

because the separation distance (proportional to resistance) is greater between corner electrodes than between the center electrode and corner electrode.

With three-phase excitation, the phase separation between adjacent corner electrodes creates a potential difference which stimulates current flow. The current flow between these electrodes generates heating along the sides of the problem domain; regions that could not be electrically heated when using single-phase excitation. The existence of additional current paths also means that less current must flow through the center electrode with three-phase excitation than with the single-phase scheme. As shown in Figure 3.8, this results in lower heating rates near the center electrode when three-phase excitation is used.

It is apparent from the plot of heating rates that heating is still more intense between the center electrode and each of the four corner electrodes than along the sides of the array between pairs of corner electrodes. This can be explained as follows. Because their phase separation is the same (120°), the magnitude of the complex potential between the center electrode and a corner electrode is the same as the magnitude of the complex potential between any two adjacent corner electrodes. However, the shorter path length (7.1 m) from center electrode to corner electrode offers less resistance to current flow than the longer path length (10 m) between corner electrodes. For the same magnitude of potential difference, more current flows along the path with the lower resistance. Thus, current flow and consequently heating intensity is greater in the region between center and corner electrodes than between adjacent corner electrodes.

For the second three-phase excitation scheme of the five-spot array

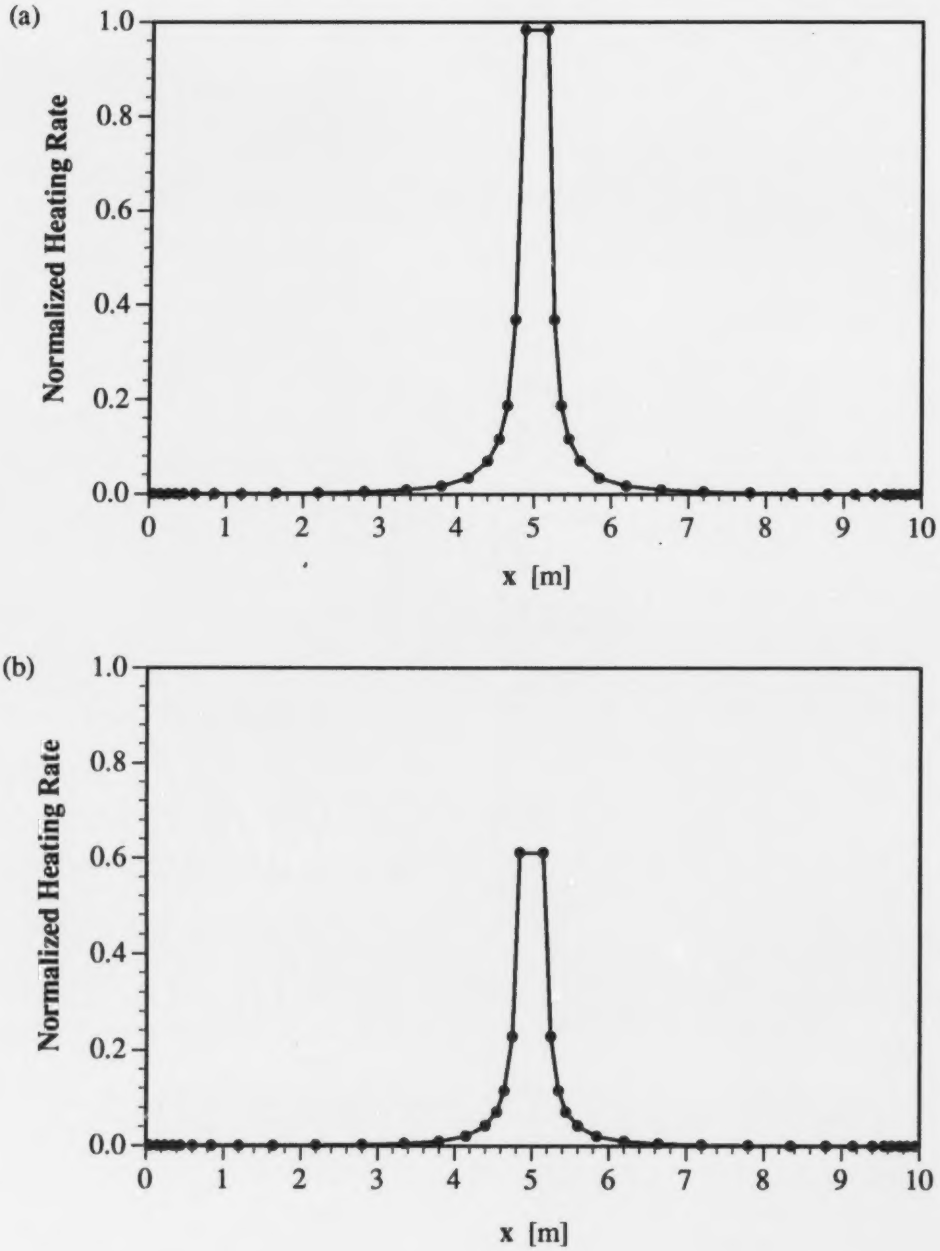


Figure 3.8 (a) Heating Rates Normalized to 1.0x10^4 W·m^{-3} Along the Plane y=5 m for Single-Phase Excitation of the Five-Spot Subarray. (b) Heating Rates Normalized to 1.0x10^4 W·m^{-3} Along the Plane y=5 m for Three-Phase Excitation of the Five-Spot Subarray.

unequal separation between the three different phases was used. Instead of having each phase separated by 120°, the separation between respective phases was 90°, 180° and 90°.

The three phase angles chosen were 0°, 90° and 270°. As with the first three-phase schemes, a phase angle of 0° was assigned to the center electrode of the five-spot pattern. Phase angles of 90° and 270° were then alternately assigned to the four corner electrodes. This phase assignment, shown in Figure 3.9(a), created a 180° phase separation between adjacent corner electrodes and a phase separation of 90° between the center electrode and each corner electrode. Like previous five-spot simulations, equal magnitudes of potential were assigned to all electrodes and a constant power input of 50 kW was specified.

The resulting distribution of heating rates is shown in Figure 3.9(b). Even though three phases were used to excite the electrodes in this case, the heating pattern is noticeably different from that produced by the previous three-phase excitation scheme (refer to Figure 3.7(b)). In general, heating is more uniform in the bulk of the formation away from the electrodes when the unequal phase separation was used. This is because the offset phasing scheme generates current flow between the center electrode and each corner electrode that is similar in magnitude to current flow between adjacent corner electrodes. With equal phase separation, the magnitude of the voltage between all electrode pairs was equal. The larger separation distance between some electrode pairs than others caused current distribution in the formation to be nonuniform. With offset phase separation, the magnitude of the potential between all adjacent pairs of electrodes is not equal. Electrodes that are out of phase by 90° would have

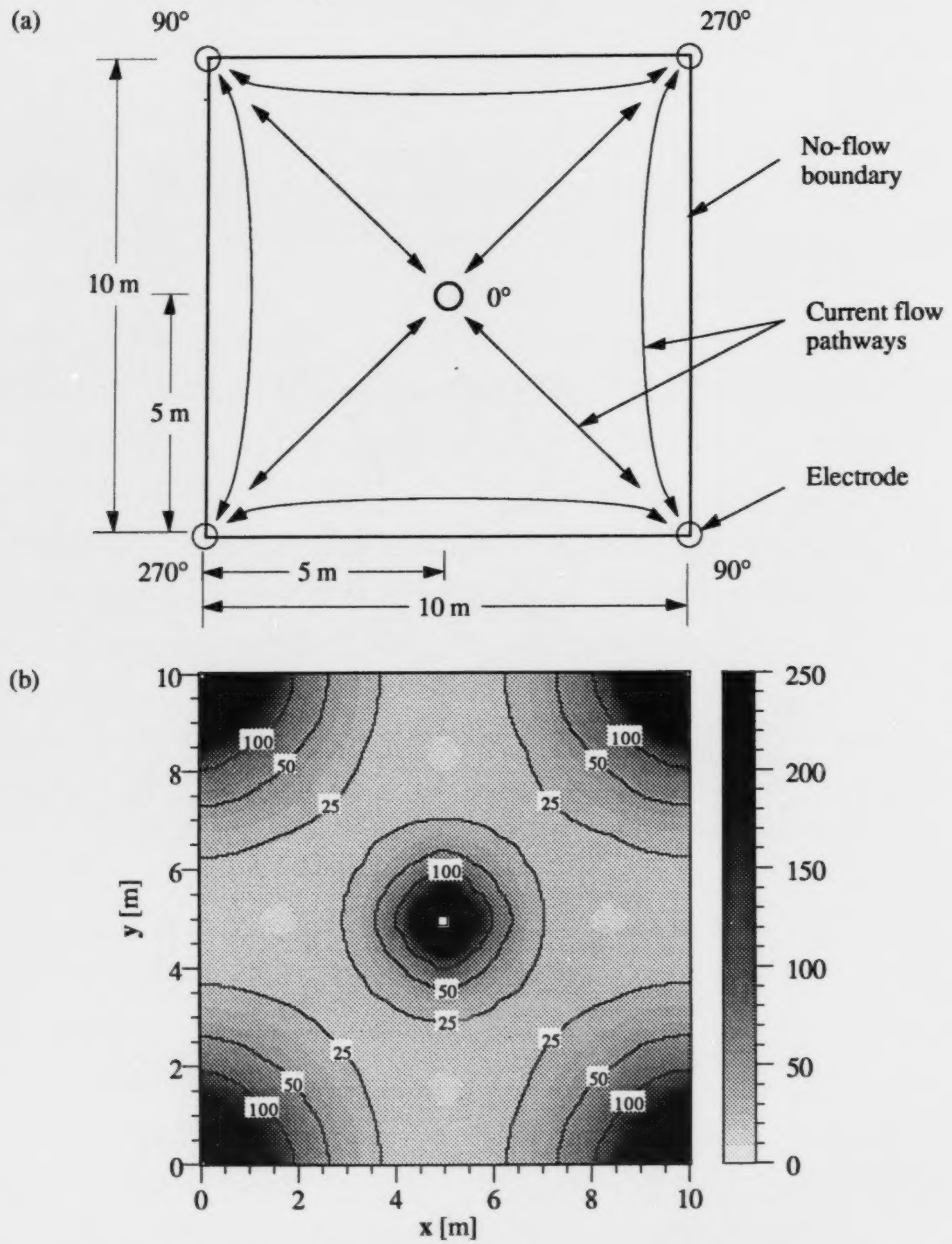


Figure 3.9 (a) Plan View of Modelled Five-Spot Subarray with Phase Angles and Current Pathways Shown for Three-Phase Excitation That Has Unequal Separation Between Phases. Phase Angles Are Given with Respect to a Common Reference. (b) Heating Rates [$W \cdot m^{-3}$] in Horizontal (x-y) Plane for Five-Spot Subarray with Three-Phase Excitation as Shown in (a). Total Input Power Is 50 kW.

a complex potential difference with a magnitude that is equal to $\sqrt{2}\,|V_e|$ where $|V_e|$ is the magnitude of the potential of each electrode in the array. Electrodes that are separated by a phase angle of 180° would have a potential difference equal to $2\,|V_e|$. For the phase assignment used then (refer to Figure 3.9(a)), a larger potential difference was created between adjacent corner electrodes than between the center and corner electrodes. This difference in potential compensated for the difference in separation distance between electrode pairs. Corner electrodes which are farther apart (10 m) possessed a current stimulating potential of $2\,|V_e|$. The more closely spaced center and corner electrodes (7.1 m) had a smaller potential difference of $\sqrt{2}\,|V_e|$. With the magnitude of potential difference between electrodes corresponding to separation distance (proportional to resistance), similar current flow along all paths was achieved. This resulted in the relatively uniform heating observed throughout the bulk of the formation. In addition, comparison of Figure 3.7(b) to Figure 3.9(b) reveals that heating intensity around the center electrode is reduced when unequal separation between the three phases is used.

The Seven-Spot Array

The seven-spot is a well placement pattern that is also commonly used in conventional oil recovery schemes. It consists of seven vertical wells drilled into a formation. When viewed from the top, six wells outline the shape of a regular hexagon and the seventh is located at the center of that hexagon.

Like the four and five-spot patterns, the seven-spot pattern is normally repeated numerous times to cover large regions of formation as shown in Figure 3.10. If considered as an electrode placement pattern that is a subarray of a large array, the seven-spot configuration would have no-flow boundaries corresponding to the sides of the regular hexagon outlined by the six outside wells. One such subarray is also indicated in Figure 3.10. Since some of the geometric planes along which these boundaries are defined would not be parallel to any of the three coordinate axes, this system of boundaries could not be accurately modelled by EPEIOS as discussed in Chapter 2. Other no-flow boundaries may be defined within the seven-spot array. However, when considering multiphase excitation, these would yield non-rectangular problem domains as well. Consequently the seven-spot configuration was studied as a single array in a formation that was, ideally, infinitely large in the plane perpendicular to the electrodes.

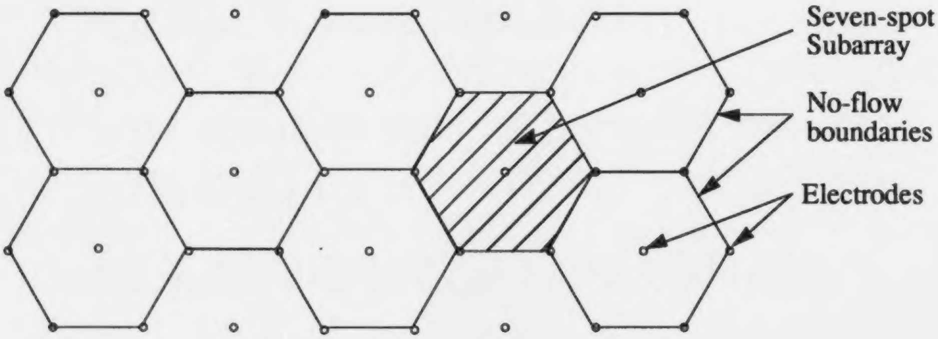


Figure 3.10 Large Array Formed From Repeated Seven-Spot Pattern.

The coordinate system of the grid structure for the seven-spot configuration was chosen such that electrodes were parallel to the z axis. As with the previous configurations, the lengths of the electrodes were made equal to the formation thickness so that the problem would be two-dimensional in x and y. An electrode spacing of 10 meters and an electrode radius of 0.2 meters were also used again.

Since a finite grid structure is used in EPEIOS to represent each problem domain, boundaries at infinity in the x-y plane had to be approximated. Testing by trial and error was performed to determine where no-flow boundaries could be defined such that they would not significantly influence current flow between electrodes in the array. It was found that sufficient separation distance between electrodes and boundaries was achieved at a distance equal to the electrode spacing. Thus a rectangular problem domain was defined for the seven-spot configuration with the four no-flow boundaries, which mark the edges of the domain, located 10 meters outside the nearest electrode in the array. The resulting dimensions of the problem space were 40 meters in the x direction and 37.4 meters in the y direction. A plan view (x-y plane) is shown in Figure 3.11(a).

As a basis for comparing single-phase and multiphase excitation, a constant power input of 100 kW was specified for both simulations. As before, the magnitude of the potential at each electrode was determined automatically by EPEIOS during the simulation.

Also shown in Figure 3.11(a) are the electrode polarities that were assigned for single-phase excitation. This electrode assignment, consisting of a positive polarity for the center electrode and negative polarities for the

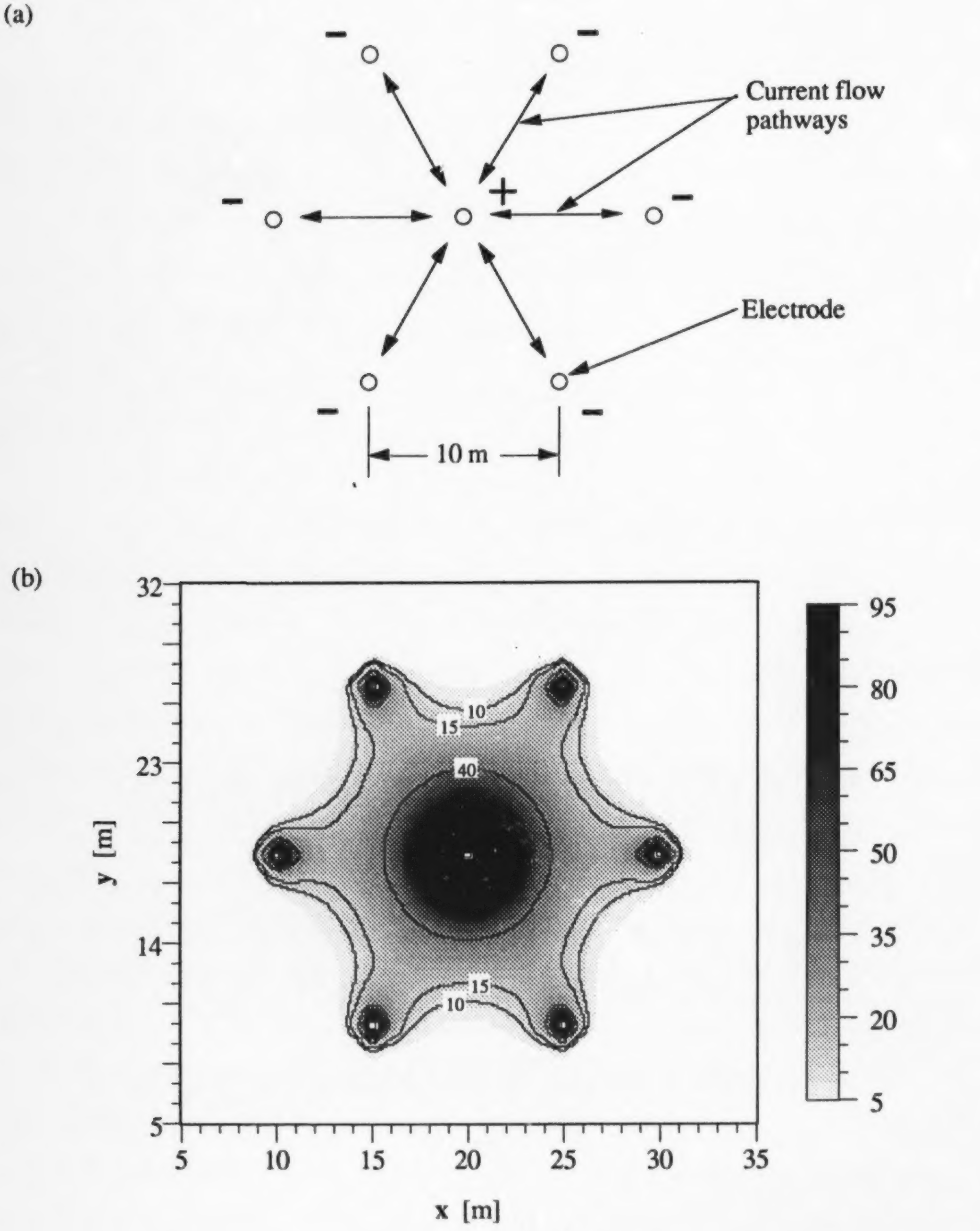


Figure 3.11 (a) Plan View of Seven-Spot Array with Polarities and Current Pathways Shown for Single-Phase Excitation. (b) Heating Rates [$W \cdot m^{-3}$] in Horizontal (x-y) Plane for Single-Phase Excitation of the Seven-Spot Array in a Problem Domain That Is Seemingly Infinite in the Horizontal Plane. Total Input Power Is 100 kW.

six surrounding electrodes, was chosen because it was a single-phase excitation scheme that had been previously proposed for the seven-spot configuration (Chute *et al.*, 1985).

Considering once again that current flow through the formation would be between electrode pairs with opposite polarity, it is apparent that six different paths of current flow would be formed by the proposed single-phase excitation. These paths would be like spokes extending from the central, positive electrode to each of the six surrounding electrodes. The involvement of the center electrode in all current flow paths implies that all current flow through the formation passes through the single positive electrode. This would create very high current densities around the central electrode relative to the outside electrodes in the array.

A greyscale plot of heating rates in the x-y plane is shown in Figure 3.11(b). To achieve greater detail near the array of electrodes, the outermost edges of the problem domain were excluded from the plot.

Examination of Figure 3.11(b) reveals that the single-phase excitation used creates channels of moderate heating between the center electrode and each of the six surrounding electrodes. There is very little heating in the regions between the outside electrodes. This result agrees with the analysis of current flow within the array.

As displayed in Figure 3.12(a), heating rates are much higher at the center electrode than at the outside ones. The very high heating rates occurring at the center electrode relative to the other electrodes would make overheating of the center electrode a problem for this excitation scheme. Overheating at the center electrode could lead to decoupling of this electrode from the formation which would cause the cessation of all

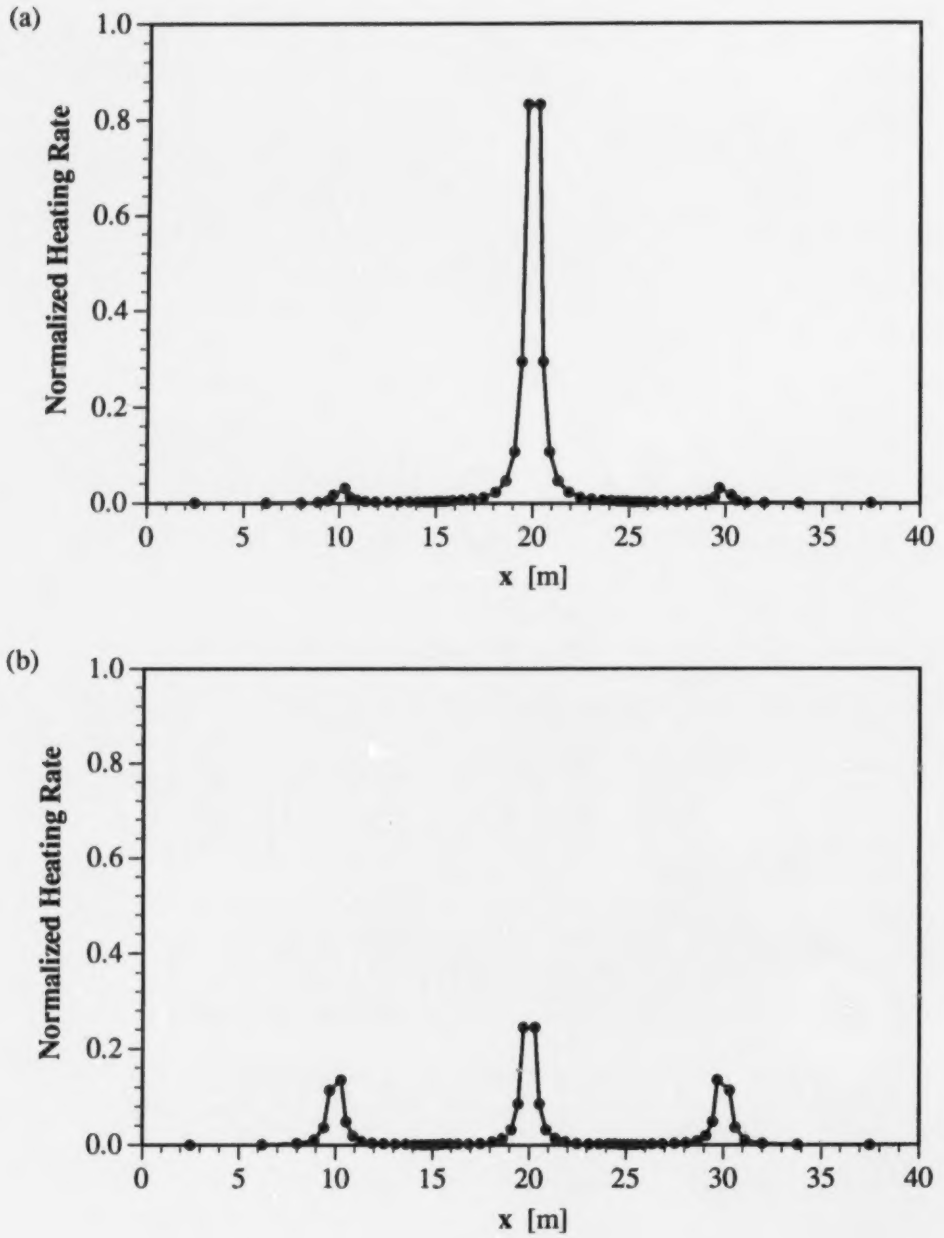


Figure 3.12 (a) Heating Rates Normalized to $1.0\text{x}10^4$ W·m^{-3} Along the Plane y=18.7 m for Single-Phase Excitation of the Seven-Spot Array. (b) Heating Rates Normalized to $1.0\text{x}10^4$ W·m^{-3} Along the Plane y=18.7 m for Three-Phase Excitation of the Seven-Spot Array.

electric heating since all the current flowing through the formation must pass through the center electrode.

Electrode potentials for three-phase excitation of the seven-spot array were assigned such that the center electrode was at one phase and the six outside electrodes alternated between the other two phases. Such an excitation scheme could be considered as six delta connections with the center electrode common to all six delta connections. The three-phase load for each delta connection would be balanced because there is equal separation between the electrodes. Phase angles for the various electrodes are shown in Figure 3.13(a). As with four-phase excitation of the square array, the magnitude of the voltage at each electrode was the same.

With three-phase excitation, a greater number of current flow pathways were generated in the formation than there were with single-phase excitation. In addition to current flow paths between the center electrode and each of the six surrounding electrodes, there would also be current flow in between the surrounding electrodes that are adjacent. The additional paths of current would enable the three-phase excitation to effect heating over a larger volume of the formation.

A greyscale plot of heating rates generated in the x-y plane of the formation by three-phase excitation of the electrodes is shown in Figure 3.13(b). As with the previous plot, the outer edges of the problem domain were excluded from this plot also.

A comparison of Figure 3.13(b) to Figure 3.11(b) shows that there are significant differences in the distribution of heating rates generated by the two different excitation schemes. With three-phase excitation, a larger volume of the formation is electrically heated than with single-phase

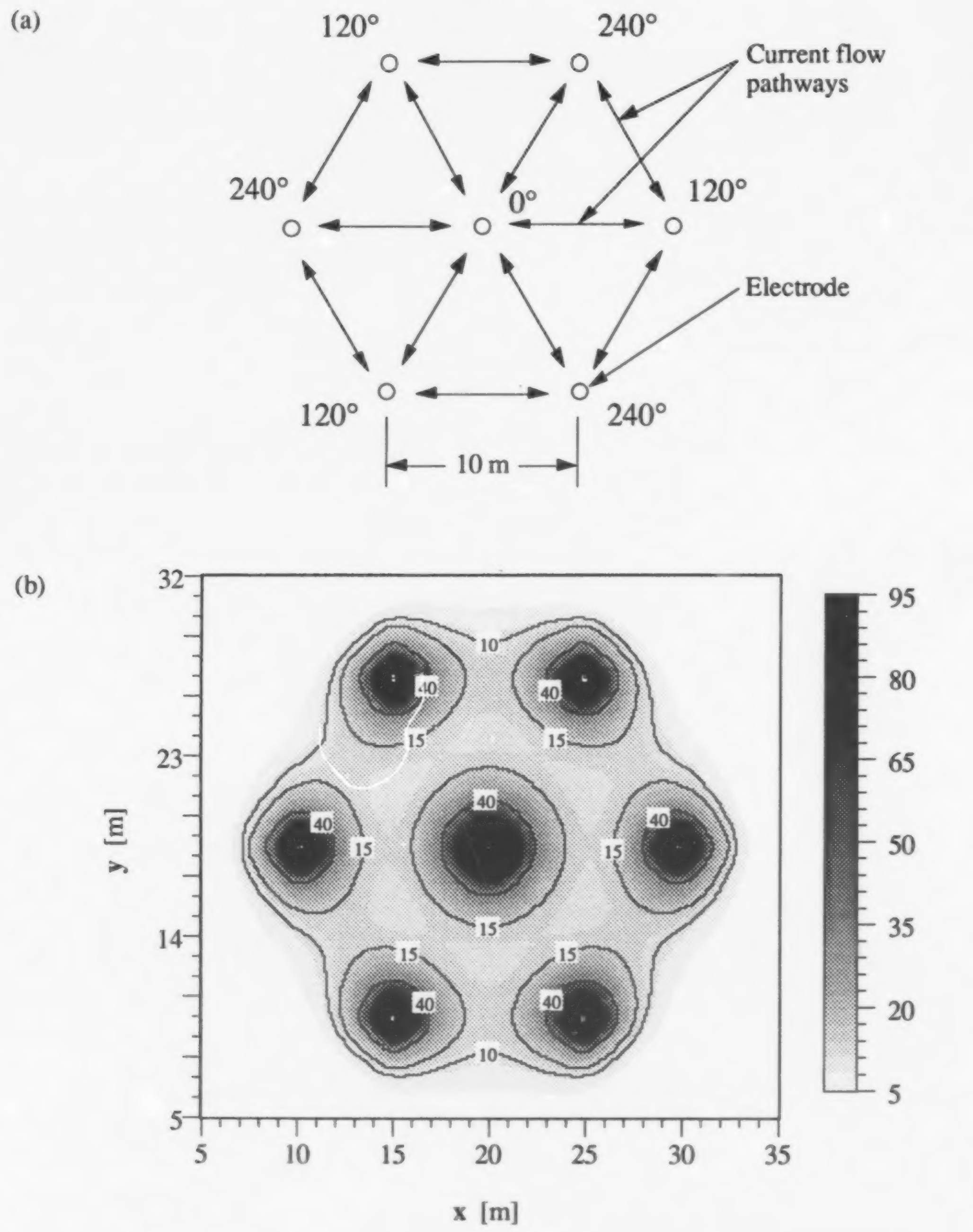


Figure 3.13 (a) Plan View of Modelled Seven-Spot Array with Phase Angles and Current Pathways Shown for Three-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Three-Phase Excitation of the Seven-Spot Array in a Problem Domain That Is Seemingly Infinite in the Horizontal Plane. Total Input Power Is 100 kW.

excitation.

Examination of Figure 3.12(b) reveals that with three-phase excitation, the heating rate at the center electrode is in fact comparable to the heating rate at the other electrodes. All heating rates seen here are substantially less than those observed at the center electrode of the seven-spot configuration when single-phase excitation was used (refer to Figure 3.12(a)). This would suggest that for the given conditions, three-phase excitation could be used to heat the formation for a longer period of time before overheating at the electrodes became a problem.

The Circular Array

To examine the possibilities of creating a zone of uniformly intense heating within an array by utilizing the properties of multiphase excitation, circular arrays involving six, eight and twelve electrodes were studied. These arrays are similar to the seven-spot array except that no center electrode is used. To facilitate the modelling of these array configurations with a rectangular grid structure, dimensions were kept relatively small. In all cases the diameter of the array was approximately 0.4 meters. However, it should be kept in mind that the actual dimensions of the array do not influence the heating patterns observed provided the relative geometry of electrode spacing and electrode size is maintained. With appropriate scaling of electrode sizes, the results obtained here may be applicable to arrays of larger dimensions such as those proposed for heating oil sand formations *in situ*.

For all simulations, electrodes were modelled as having a diameter of 2 centimeters and a length of 10 centimeters. Input power to each array was kept constant at 50 W. Like other simulations involving a vertical array of electrodes, the thickness of the heated material was specified as being equal to the length of the electrodes in order to keep the problem two dimensional. Also like other simulations, the coordinate axes for these simulations were chosen such that the electrodes were parallel to the z axis.

Each simulation was conducted as if the array were located in a problem space that was seemingly infinite in the horizontal (x-y) plane. As with previous simulations for the seven-spot array, the infinite domain was approximated by defining rectangular no-flow boundaries around the array at a large distance from the electrodes relative to the electrode spacing. By trial and error it was determined that a sufficient distance between the electrodes and the no-flow boundaries was approximately 1 meter.

The isotropic material to be heated was characterized by arbitrarily chosen values for electrical conductivity, thermal conductivity and thermal capacity, as input of these values is required by EPEIOS. As discussed in Chapter 2, all patterns of heating rates produced are independent of these values.

The first circular array that was examined was one consisting of six electrodes. These electrodes were symmetrically positioned in the material such that when viewed from the ends they marked out the points of a regular hexagon.

For multiphase excitation of this array, six different phases were used. The phase angles chosen were 0°, 60°, 120°, 180°, 240° and 300°. Phases were assigned around the array in order of increasing phase angle as shown in Figure 3.14(a). In this manner, all adjacent electrodes in the array are separated by the same phase difference (60°) and opposing

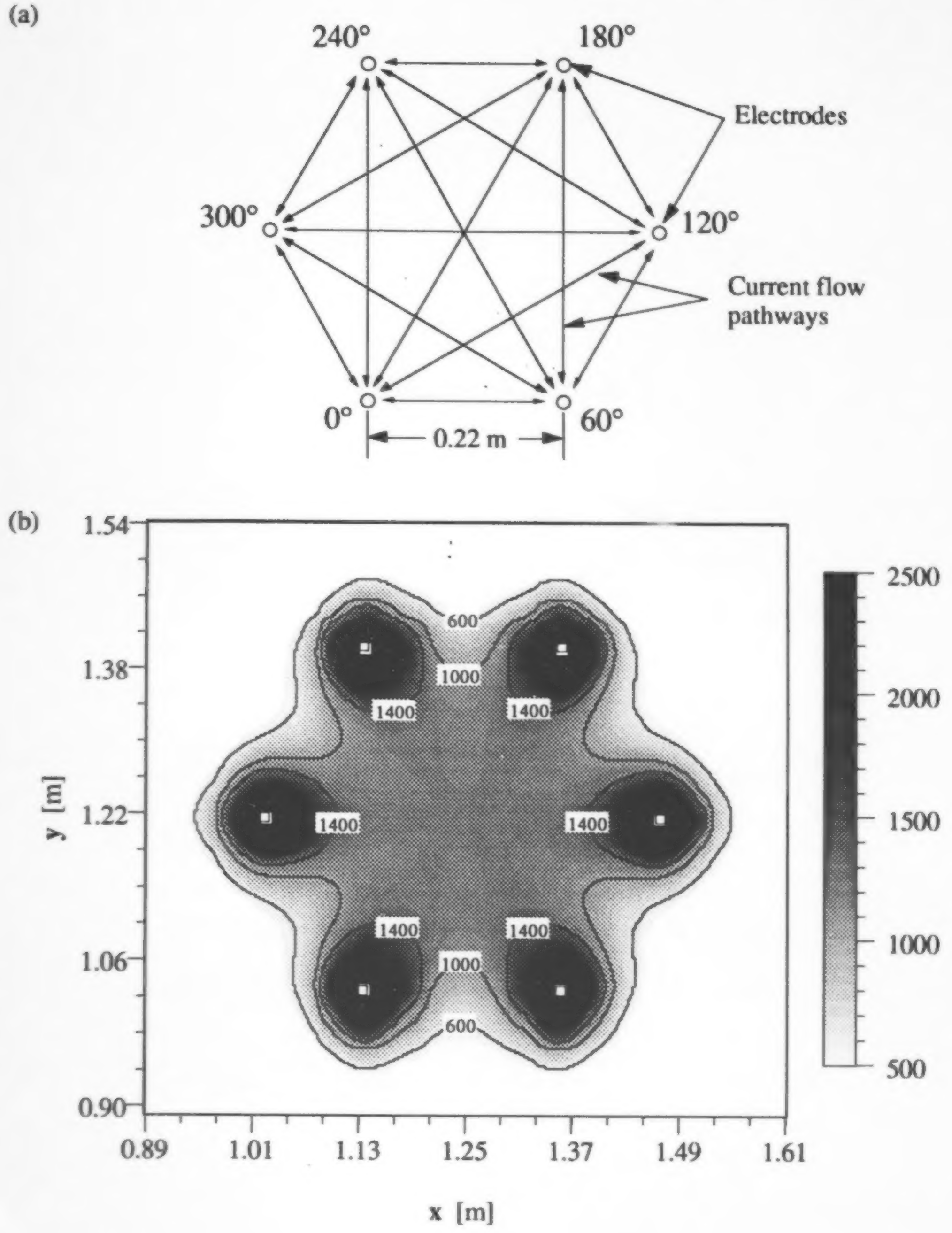


Figure 3.14 (a) Plan View of Modelled Hexagonal Array with Phase Angles and Current Pathways Shown for Six-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Hexagonal Array with Six-Phase Excitation as Shown in (a). Total Input Power Is 50 W.

electrode pairs had a phase separation of 180°.

A greyscale plot of the distribution of heating rates in the near vicinity of the array is shown in Figure 3.14(b). Notice the greyscale for the plot which indicates that a window of heating rates between 500 $W \cdot m^{-3}$ and 2500 $W \cdot m^{-3}$ is displayed. This scale was chosen to most clearly show the distribution of heating rates within the confines of the array which is the area of interest.

It is readily apparent from the plot of heating rates that heating is very uniform inside of the array away from the electrodes. The uniformity of heating rates suggests that with this configuration, there are no dominant current flow pathways.

To understand these results, it must first be realized that with each electrode in the array being excited at a unique phase angle, each electrode is effectively at a different potential than the rest. Consequently there exists a potential difference between all electrode pairs in the array. The magnitude of the potential difference between each electrode pair is, of course, dependent on their phase separation. The smaller the phase separation is, the smaller the potential difference will be.

With a complex potential difference between each electrode pair in the array, current flow would be stimulated between each pair of electrodes. These current flow pathways are indicated in Figure 3.14(a). The magnitude of current flow between any two electrodes is determined by the magnitude of the potential difference and the effective impedance (assumed real) seen by the current. The effective impedance is dependent upon the separation distance between electrodes.

For the six-phase excitation of the hexagon array, the magnitude of the potential difference between electrode pairs is somewhat proportional to the impedance between them. Adjacent electrode pairs exhibit a relatively small phase separation and thus a small potential difference but they are closely situated. Opposing electrodes have a relatively large phase separation but they are farther apart. As a result of the proportionality of potential difference and impedance, the magnitude of current flow between all the various electrode pairs is comparable. Consequently, heating throughout the formation inside of the array is quite uniform. This is the same effect that was observed with four-phase excitation of the square array.

For the sake of comparison, single-phase excitation of the same circular array of six electrodes was simulated. As shown in Figure 3.15(a), voltage potentials were assigned to the electrodes such that positive and negative polarities alternated around the circumference of the array. A constant input power of 50 W was maintained.

A greyscale plot of the resulting distribution of heating rates is shown in Figure 3.15(b). It is apparent from the plot that much less heating occurs at the center of the array with this single-phase scheme than with the multiphase scheme. Here, most of the heating occurs in the regions between adjacent electrodes. This result was expected since the potential difference between all electrode pairs of differing polarity is the same with single-phase excitation. Although current paths would exist between opposing electrode pairs, the distance between them is much greater than the distance between adjacent electrodes. For equal potential difference, most of the current flow will occur along the path of least resistance (the shortest path) which is the one between adjacent electrodes.

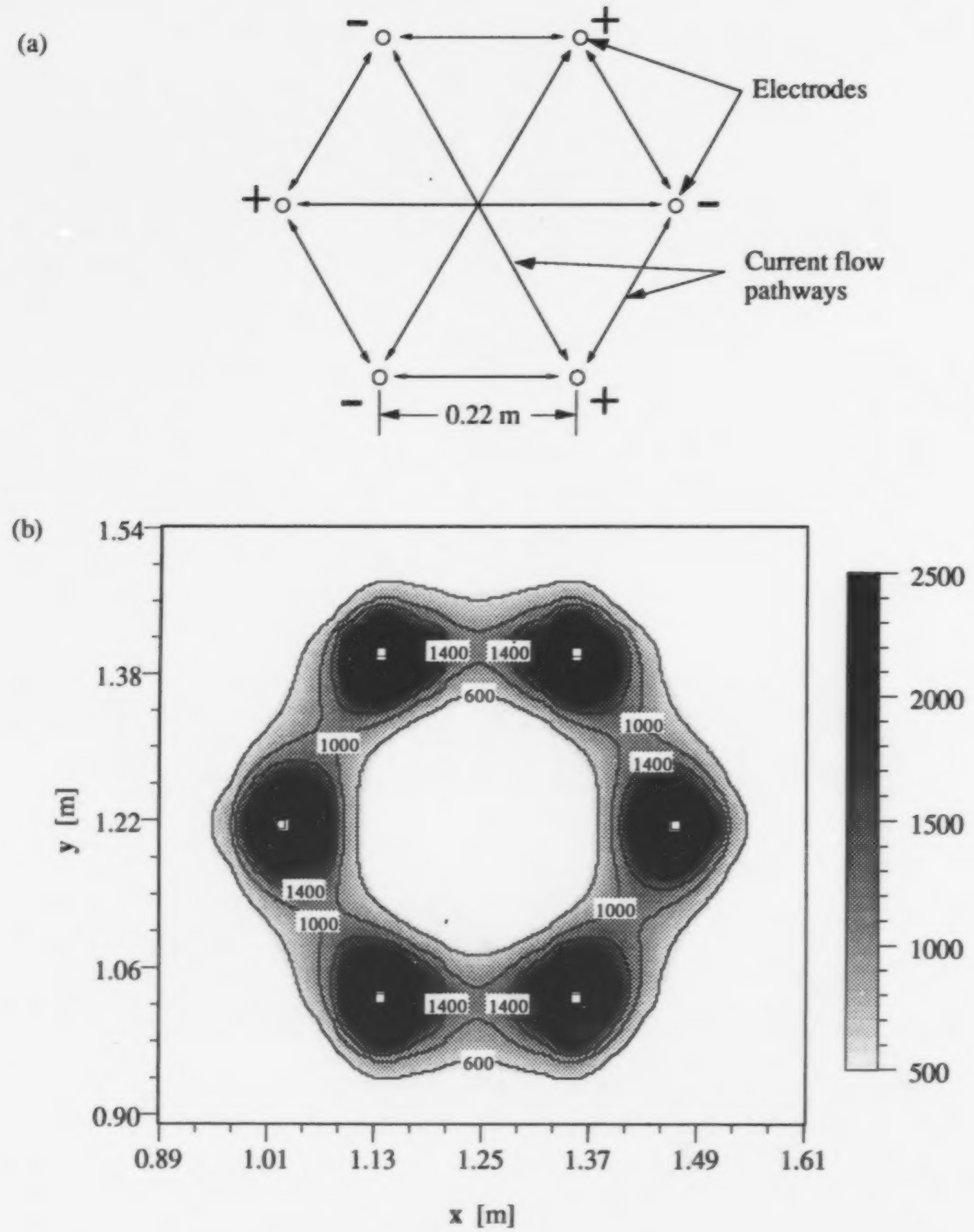


Figure 3.15 (a) Plan View of Modelled Hexagonal Array with Polarities and Current Pathways Shown for Single-Phase Excitation. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Hexagonal Array with Single-Phase Excitation as Shown in (a). Total Input Power Is 50 W.

Similar to the six electrode array, the eight electrode array consisted of eight electrodes equally spaced in a circle as shown in Figure 3.16(a). Each electrode was again given a unique phase with phase angles of 0°, 45°, 90°, ..., 315° being assigned sequentially around the array. Also shown in Figure 3.16(a) are the current flow pathways that are generated by eight-phase excitation of the eight electrode array. It is apparent that the number of pathways is significantly greater for the eight electrode configuration than for six-phase excitation of the six electrode array (refer to Figure 3.14(a)).

The distribution of heating rates that was generated by eight-phase excitation of the eight electrode array is shown in Figure 3.16(b). As with six-phase excitation of the six electrode array, heating rates are very uniform within the array. However, heating rates in the material enclosed by the eight electrode array are slightly higher on average. This can be attributed to the larger number of current paths that exist with eight electrodes and eight phases. Also with eight electrodes instead of six, heating right around each of the electrodes is less intense. With the greater number of electrodes and phases, current flow through each electrode is reduced thereby reducing current densities and heating rates in the near vicinity of the electrodes.

Increasing the number of electrodes in the circular array to twelve yielded the configuration in Figure 3.17(a). For twelve-phase excitation, phase angles of 0°, 30°, 60°, ... , 330° were used. Once again these phases were assigned one to each electrode sequentially around the array. With this configuration, a very large number of current paths are generated within the array. These current paths are also indicated in Figure 3.17(a).

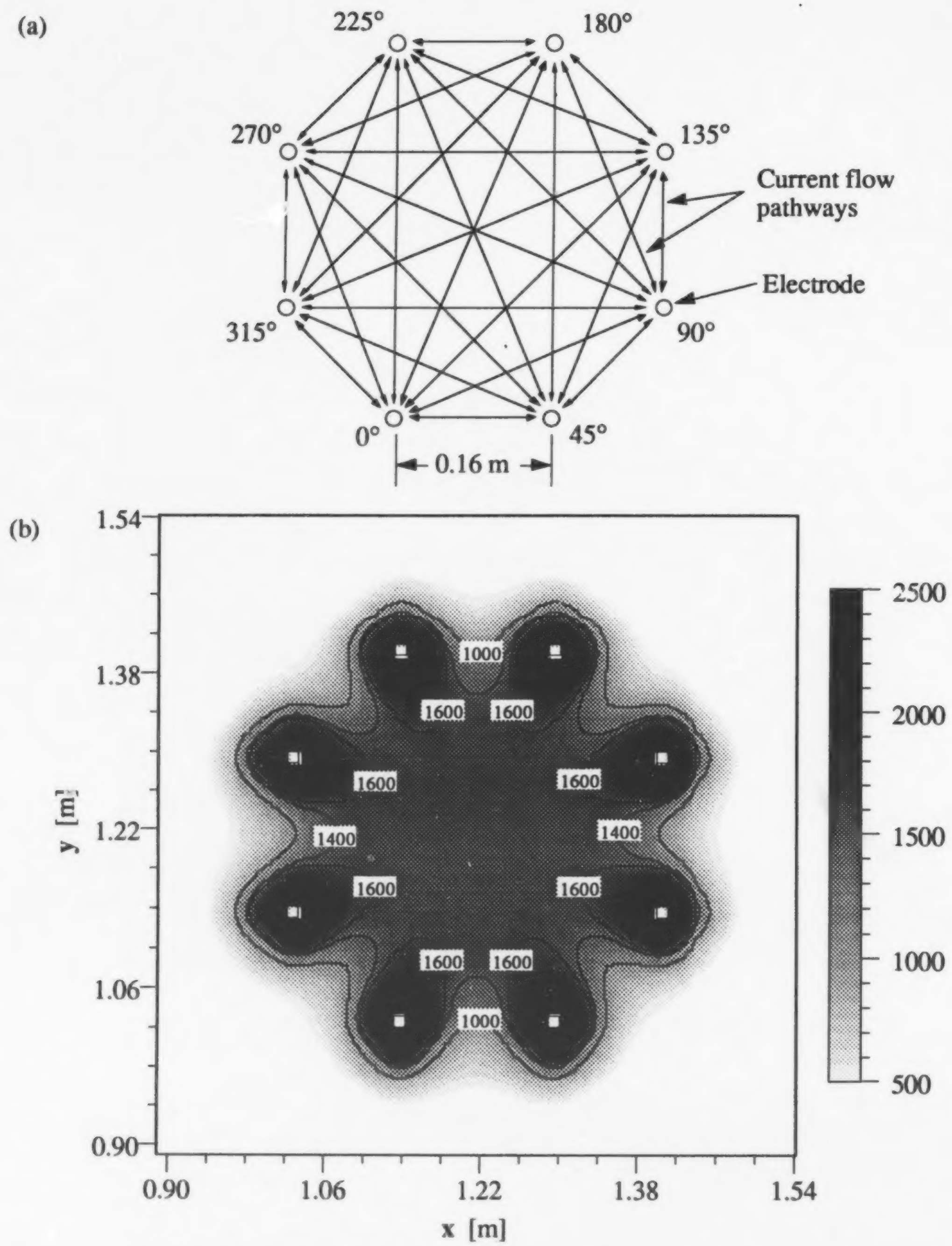


Figure 3.16 (a) Plan View of Modelled Octagonal Array with Phase Angles and Current Pathways Shown for Eight-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Octagonal Array with Eight-Phase Excitation as Shown in (a). Total Input Power Is 50 W.

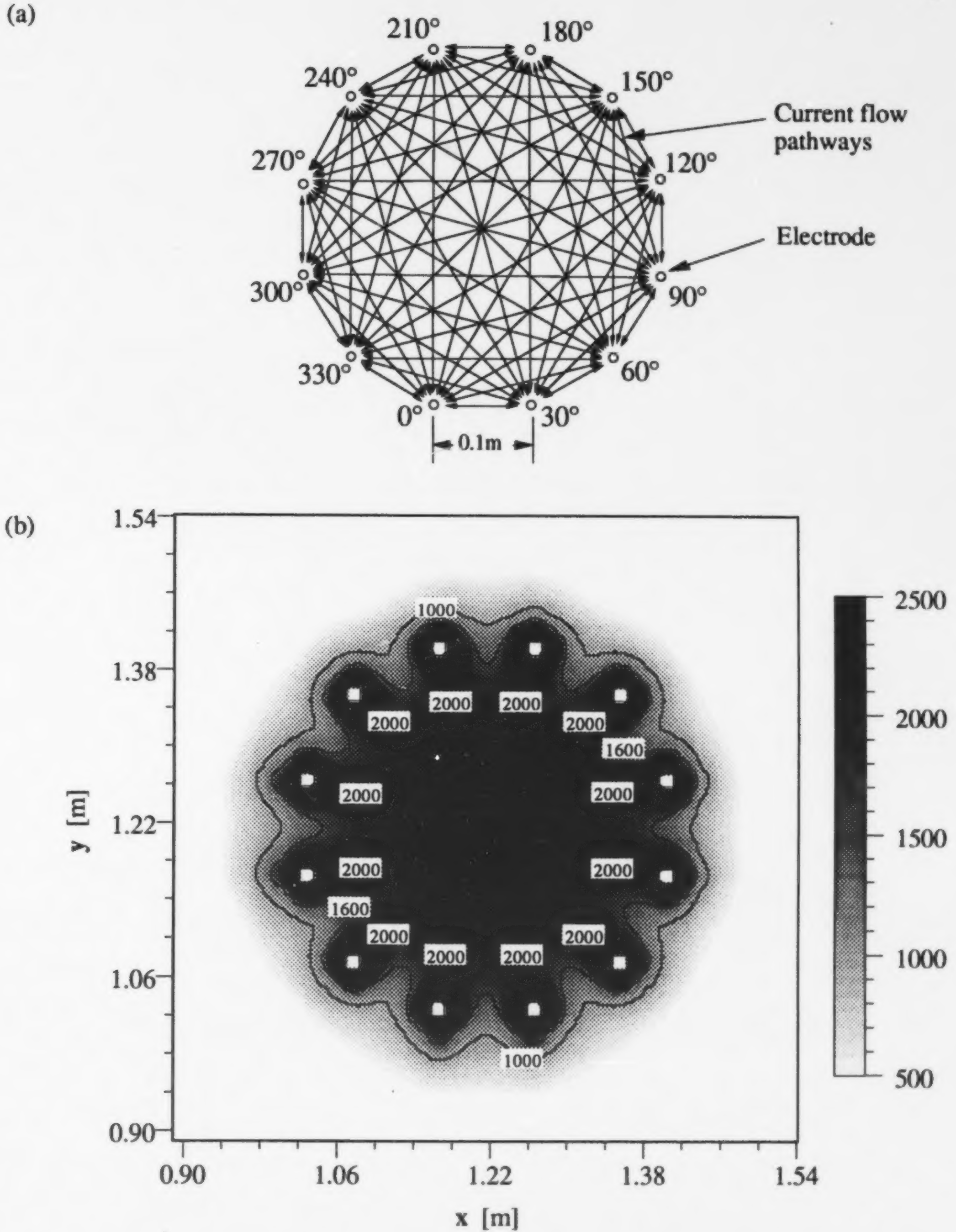


Figure 3.17 (a) Plan View of Modelled Dodecagonal Array with Phase Angles and Current Pathways Shown for Twelve-Phase Excitation. Phase Angles Are Given with Respect to a Common Reference. (b) Heating Rates [W·m^{-3}] in Horizontal (x-y) Plane for Dodecagonal Array with Twelve-Phase Excitation as Shown in (a). Total Input Power Is 50 W.

A greyscale plot of the heating rates generated by the twelve electrode array is shown in Figure 3.17(b). Like the circular array of six and eight electrodes, heating rates were once again very uniform within the boundaries of the array. However, as seen before when the number of electrodes and phases were increased from six to eight, increasing the number of electrodes from eight to twelve resulted in an increase in the average heating rates within the circular array. In addition, heating intensity right around each electrode in the twelve electrode array is lower than the eight electrode array. The reasons for these differences would be the same as those mentioned in comparison of the eight electrode array to the six electrode array.

To clarify the differences in heating patterns within the various circular array configurations that were examined, heating rates were plotted for each array along a vertical (y-z) plane which passes through the center of the array. For the six electrode configurations, the y-z plane corresponded to the plane x = 1.25 m. For the eight and twelve electrode configurations, the y-z plane corresponded to the plane x = 1.215 m. In all cases, the chosen plane passed between adjacent electrodes on opposing sides of the array.

The resulting graph of normalized heating rates for the four different circular array configurations is shown in Figure 3.18. From this graph it is clear that there is a vast difference between heating generated by six-phase excitation and heating generated by single-phase excitation of the six electrode array. With single-phase excitation, there is virtually no heating in the interior region of the array. Most of the heating is confined to the edges of the array. With six-phase excitation, uniform heating is generated

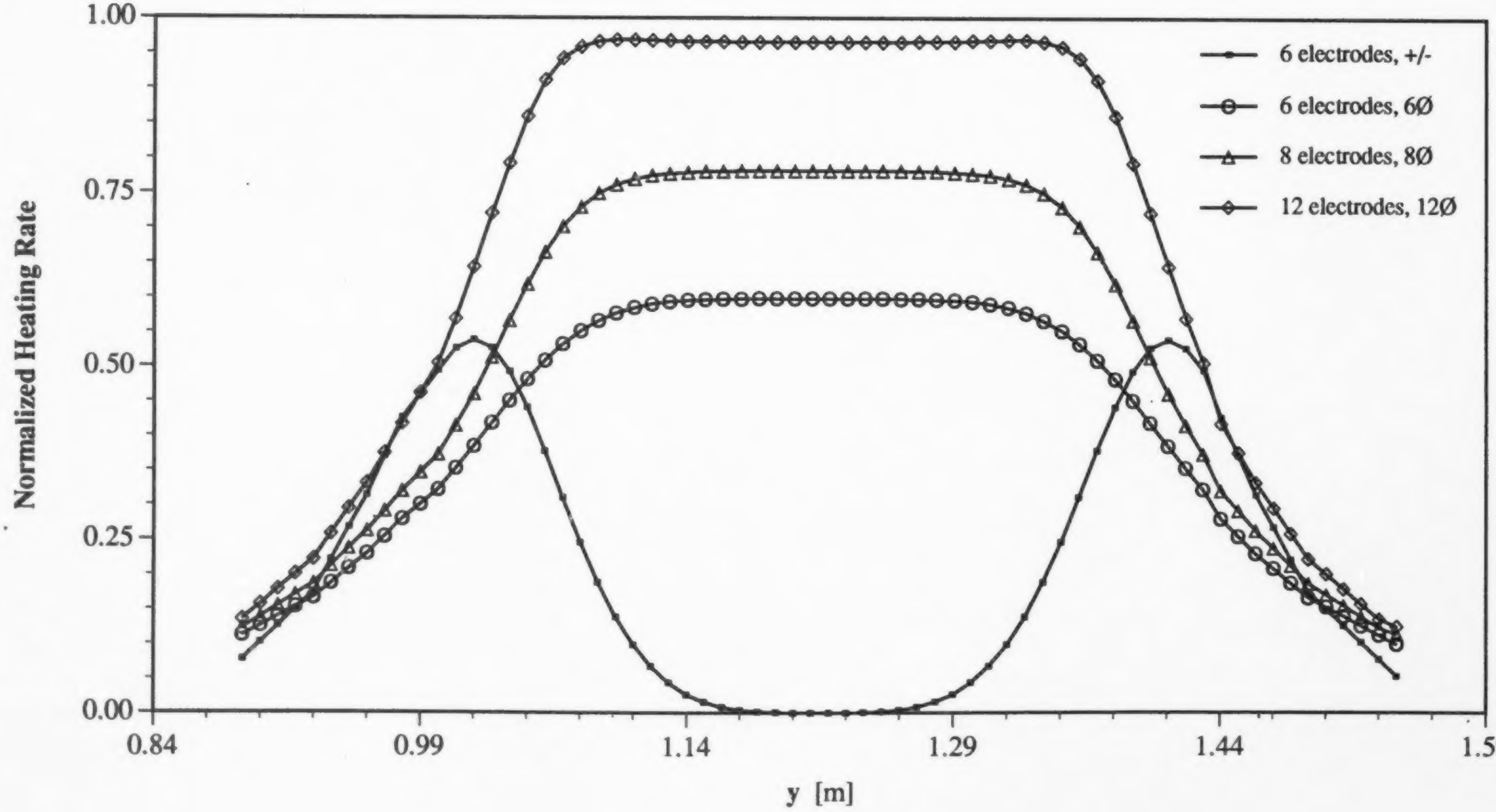


Figure 3.18 Heating Rates for the Different Circular Arrays Taken Along the Vertical (y-z) Plane Which Passes Through the Center of the Array. Heating Rate Values are Normalized to 2 x 10^3 W·m^{-3}.

throughout most of the region within the array. This effectively creates a hot zone within the confines of the array. The intensity of this heating is slightly greater than that seen between adjacent electrodes when single-phase excitation is used. The graph also shows that as the number of electrodes and phases are increased, the intensity of the heating within the array increases significantly. In addition, the use of more electrodes and phases results in uniform heating over a greater portion of the region enclosed by the electrodes. It is expected that continuous variation of phase around a cylindrical region would generate completely uniform heating within that region.

The Single Well Configuration

When modelling the heating of underground formations with the single well configuration it was assumed that many wells in some form of closely spaced array would be used. For convenience the array pattern was chosen to be a square grid with wells placed in equally spaced rows and columns. The plan view of such a well placement is shown in Figure 3.19.

As with previous electrode arrays, vertical planes of symmetry which can be modelled as no-flow boundaries may be defined. With each well having the same electrode configuration, vertical planes of symmetry exist half way between each row and column of wells. When viewed from above, these planes outline a square problem domain around each well. The solution within each such problem domain would be identical and thus a solution for the entire array could be obtained. The square problem domain around each well may, however, be further reduced by considering that other planes of symmetry exist along each row and column of wells. These additional planes of symmetry divide the square region containing each well into four quadrants. Each quadrant would contain one quarter of the total volume of the well and the solution within each quadrant would also be identical. Such a quadrant, surrounded by no-flow boundaries, was used as the problem domain for this study of the single well configuration. The planes of symmetry for the array of single well electrode configurations and the problem domain used in the simulations are also indicated in Figure 3.19. It was assumed that the wells in the array were spaced 40 meters apart. This meant that the problem domain was 20 meters by 20 meters in the horizontal plane (x-y plane).

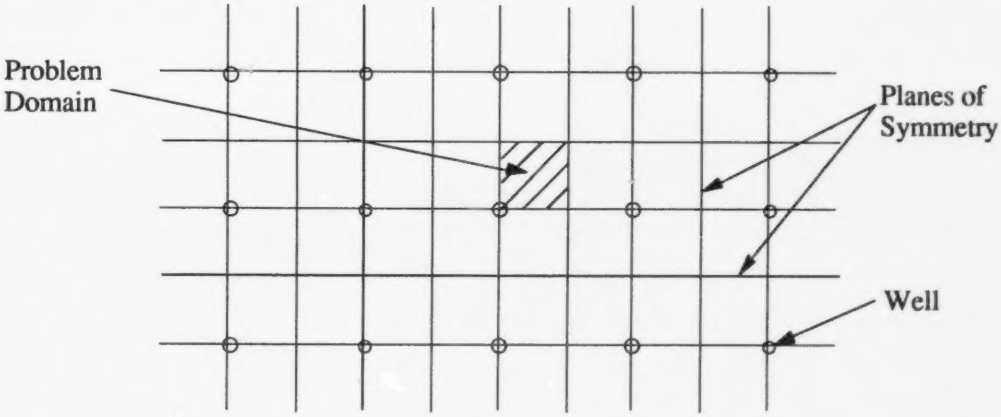


Figure 3.19 Plan View of an Array of Single Wells and Corresponding Planes of Symmetry.

Unlike previous simulations, a problem domain that was non-homogeneous in the vertical (z) dimension was considered for the single well configuration. With the top/bottom arrangement (a single electrode with current return through the upper casing), current flows from the bottom electrode up through the formation to the well casing. As a result, there is current flow through the earth material that is above the layer of oil bearing formation. Typically, this overburden possesses a much higher electrical conductivity than the layer containing oil. Thus, to achieve greater accuracy in the simulation, the problem domain was divided into two layers. The top layer, which represents the overburden, was assigned an electrical conductivity that was ten times greater than that of the bottom layer which represents the oil formation.

The thickness of the overburden was modelled as being seemingly infinite since oil formations heated by a single well would typically be deeply buried. Similar to previous simulations, an approximation was obtained by increasing the thickness of the overburden until the solution near the well section where heating occurs was virtually unaffected by a further increase in thickness. For generality, the thickness of the oil formation was assumed to be much greater than the length of the section of well bore around which heating occurs.

When heating an underground formation around a single well, it is desirable to heat along the section of the well that falls within the formation stratum that contains oil. The length of the section may vary and it will depend somewhat on the thickness of the oil rich layer. For this investigation, the length of the section of well over which heating was to occur was chosen to be 20 meters.

In modelling the top/bottom configuration then, the bottom electrode region of the well was specified as being 20 meters in length. This electrode region was separated from the remainder of the well by a 5 meter section of well bore that was both thermally and electrically insulating. The radius of the electrode section, the insulated section and the upper casing was all the same and equal to 0.2 meters.

As shown in Figure 3.20, the well was positioned in the problem domain such that the interface between the oil bearing layer and the overburden was at the center of the 5 meter long insulated section of well bore. The thickness of oil bearing formation was specified as being 32.5 meters. It extended 10 meters below the bottom of the electrode. This was found to be sufficient to model an oil deposit that was much thicker than the length of the electrode. The overburden was modelled as being 127.5 meters thick. This meant that the well casing which extends from the top of the insulated section of well bore to the top of the problem domain was 125 meters long.

Exciting the top/bottom electrode configuration with constant input power of 40 kW (10 kW in modelled problem domain) yielded the distribution of heating rates displayed in Figure 3.21. This plot shows the variation of heating rates along the length of the electrode and a portion of the casing out to a radial distance of approximately 4 meters from the center of the well. As expected, slight hot spots are evident very near the electrode at the two ends. These are a result of the higher current densities which naturally exist at the ends of a finite electrode. The hot spot at the top of the electrode is slightly more intense because this end of the electrode is closer to the casing return and thus more current will leave at this point.

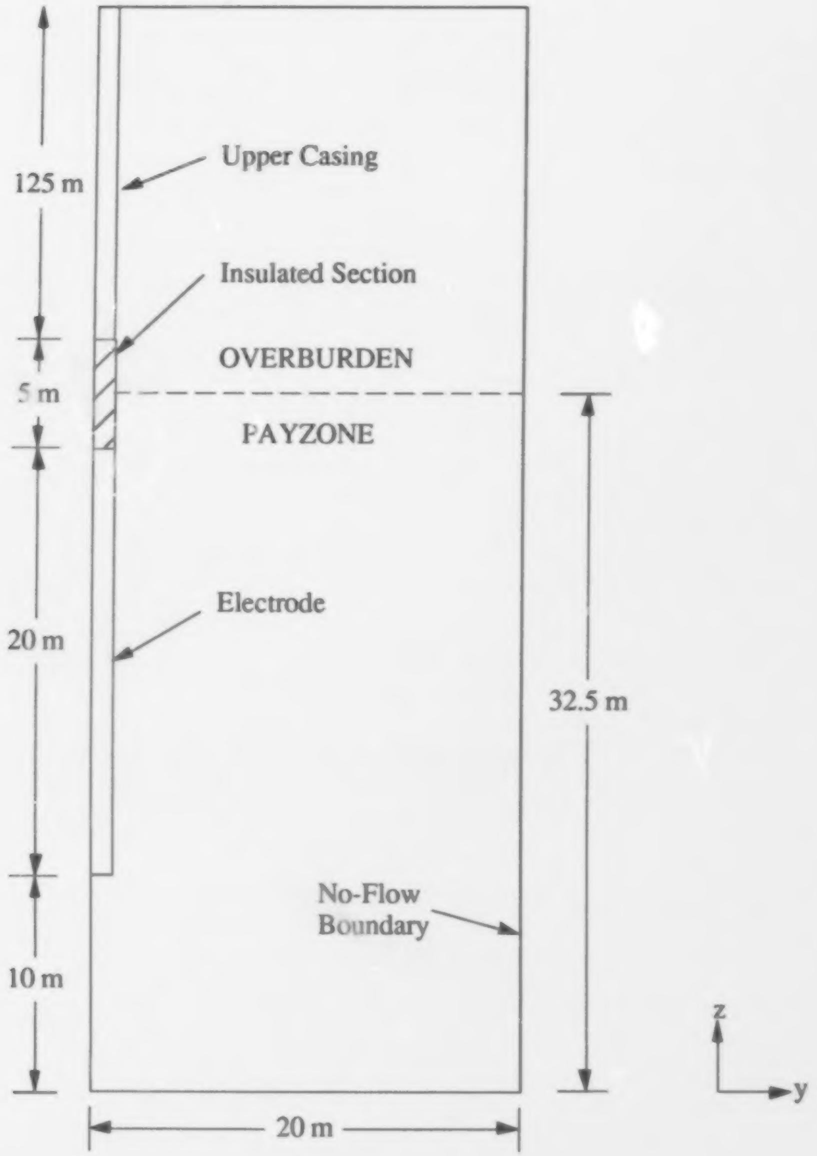


Figure 3.20 Problem Domain for Top/Bottom Configuration.

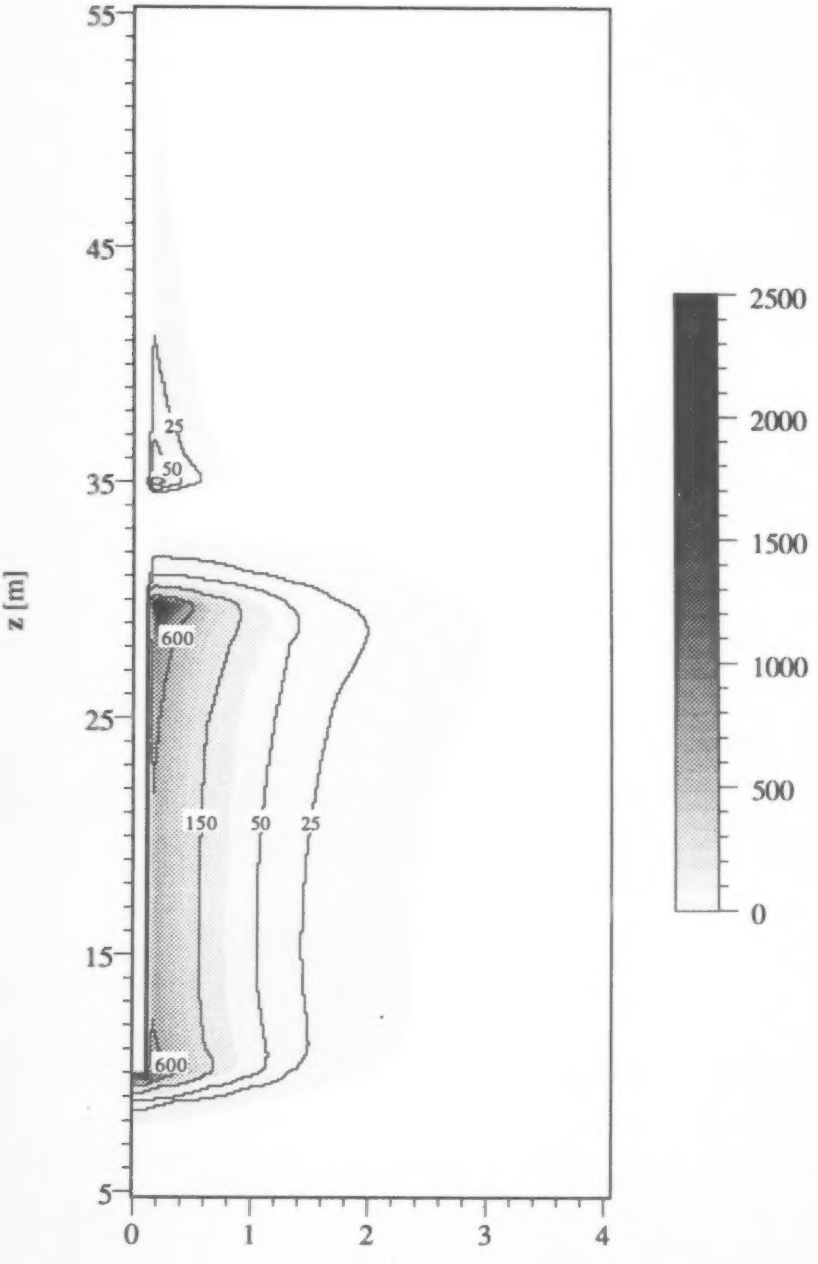


Figure 3.21 Heating Rates [W·m^{-3}] in Vertical (y-z) Plane for Single-Phase Excitation of Top/Bottom Configuration Shown in Figure 3.20. Formation Is Layered (Interface at z = 32.5 m) with Electrical Conductivity Approximately Ten Times Greater in Top Layer Than in Bottom Layer. Total Input Power Was 40 kW.

This results in non-uniform heating along the length of the electrode. There is also indication of significant heating along a portion of the upper well casing. Heating in this region is undesirable since it occurs in the overburden where there is no recoverable oil.

To better understand the heating that is generated by the top/bottom configuration, a second simulation was performed. This simulation involved driving the electrodes with a constant input power of 40 kW for an extended period of time until a maximum temperature of 150°C was reached. The starting temperature for the entire problem domain was specified as being 10°C. The same thermal properties were used for both layers in the problem domain since thermal properties do not vary too much for the different types of earth material that are typically found near an oil reserve (Stroemich *et al,* in press). The actual values that were used were a thermal conductivity of $1.65\ \mathrm{W \cdot m^{-1} \cdot K^{-1}}$ and a thermal capacity of $2.04 \cdot 10^{6}\ \mathrm{J \cdot m^{-3} \cdot K^{-1}}$. These values are representative of rich Athabasca oil sand (Stroemich *et al,* in press). For this simulation, the thermal dependence of the electrical conductivity of both layers in the problem domain had to also be specified. The bottom layer was assigned a temperature dependence for electrical conductivity that is associated with rich Athabasca oil sand. The top layer was assigned a temperature dependence for electrical conductivity that approximated that of a shale like material. The thermal dependencies of the electrical conductivity for the two layers were chosen somewhat arbitrarily. However, the results of the simulations should still be accurate since, like the thermal properties themselves, the temperature dependence of electrical conductivity is very similar for all earth materials that are typically found near an oil reserve for the temperature range used in this simulation (Stroemich *et al*, in press).

The time duration over which heating occurred was approximately 174.5 hours. The resulting temperature distribution is shown in Figure 3.22.

As expected, given the heating rate distribution generated by this configuration (refer to Figure 3.21), regions of higher temperatures can be identified at the two ends of the electrode. However, temperatures at the top end of the electrode are much higher than at the bottom end (more than two times). It was not immediately obvious from the plot of heating rates that temperatures would be so much higher at the top of the electrode than elsewhere along the electrode. However, this result may be explained as follows. As heating progresses and the temperature of the formation increases, the electrical conductivity of the formation increases accordingly. At the top of the electrode where heating rates are highest, the temperature and the electrical conductivity of the formation would increase most rapidly. The increase in conductivity of the formation around the top of the electrode reduces the resistance between the electrode and the upper casing of the well. The reduced resistance causes more current to flow through the electrode at this point which in turn causes an increase in heating rates. This contributes to a more rapid increase in formation temperature and electrical conductivity and a continued increase in current flow and heating rates. In this manner, heating becomes focussed at the top of the electrode and a very non-uniform temperature distribution is created. Such a temperature distribution would limit oil recovery since a significant temperature increase is achieved along only a few meters of the electrode.

For the multiple electrode configurations with multiphase excitation, three electrode segments excited at three different phase angles were used.

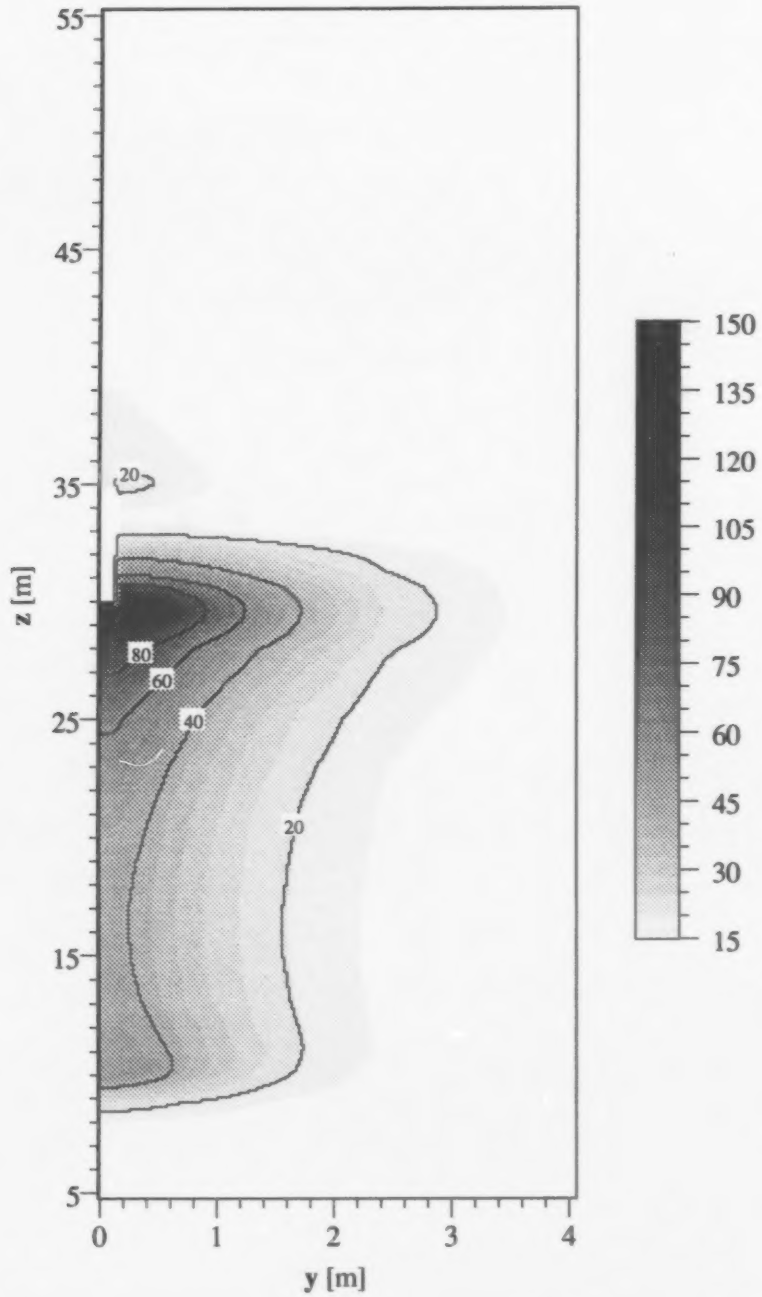


Figure 3.22 Temperatures [°C] in Vertical (y-z) Plane for Single-Phase Excitation of Top/Bottom Configuration Shown in Figure 3.20 After Heating From an Initial Temp. of 10°C to a Maximum Temp. of 150°C. Formation Is Layered (Interface at $z = 32.5$ m) with Electrical Conductivity in Top Layer Approximately Ten Times That in Bottom Layer. Input Power Was a Constant 40 kW and Total Heating Time Was 174.5 Hours.

Three-phase excitation was chosen here because it would be a practical choice for actual implementation schemes. It was also felt that three phases would be sufficient to demonstrate the significant properties of multiphase excitation used in this manner.

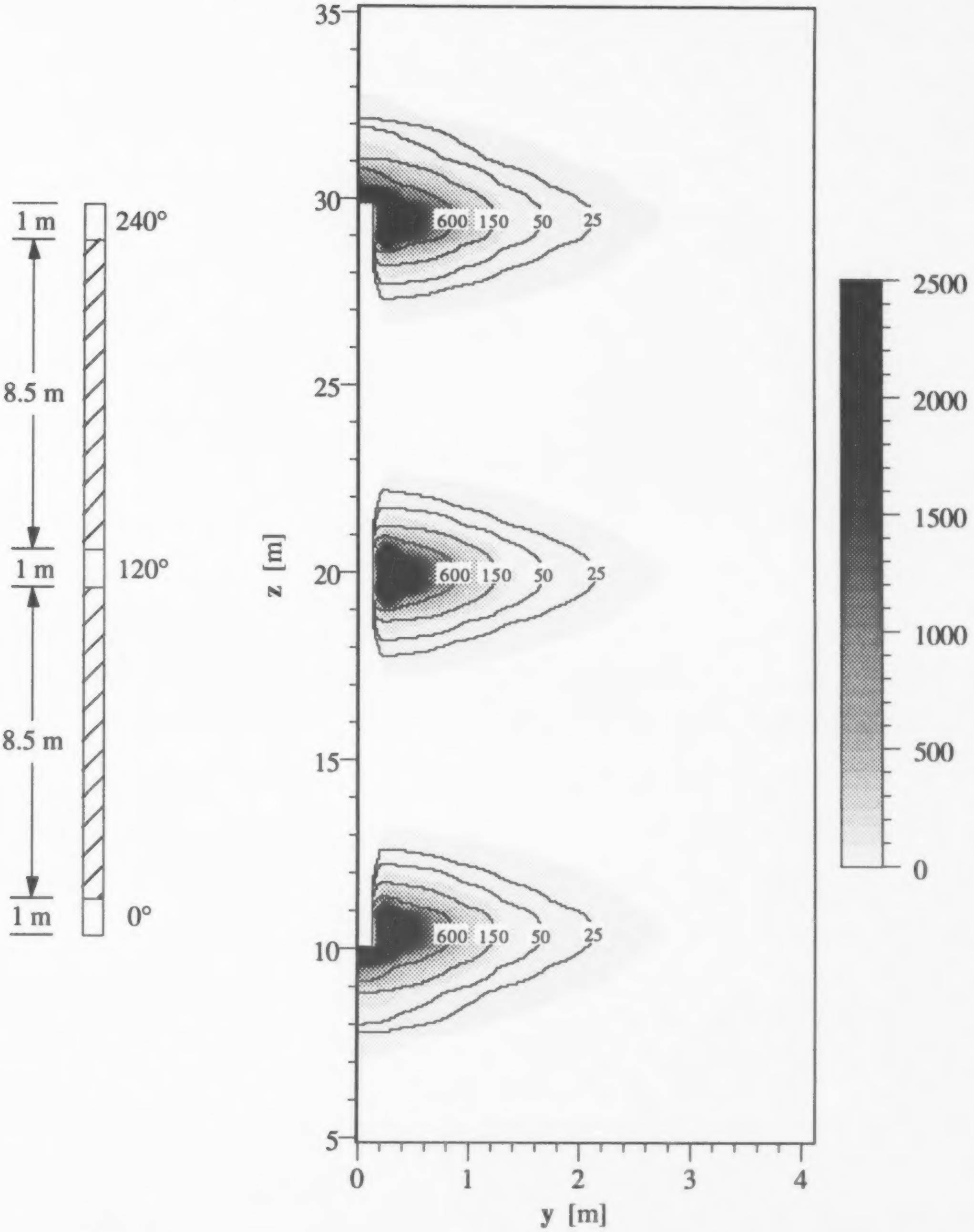
The analysis of the type of heating patterns generated by three electrodes with three-phase excitation began with an investigation into the influence of electrode length and separation on the distribution of heat. Initially, it was unclear if short electrodes separated by comparatively long insulated regions would yield a more uniform distribution of heating along the length of the well section than larger electrodes separated by comparatively short insulated regions. To investigate, a series of simulations were performed. For each simulation, the length of the three electrodes and the length of the two insulated regions between them were different. To isolate the effects of electrode length and spacing, the combined length of the electrode and insulated spacings was kept constant at 20 meters. The well bore above the 20 meter section was not included in these simulations. In addition, the same multiphase excitation scheme was consistently used. In all cases, electrodes were excited with regular three-phase power so that each electrode had a unique phase angle and the phase separation between all electrodes was the same (120°). The input power to these electrodes was also kept constant at 40 kW.

For simplicity, these initial simulations involving multiple electrodes were conducted as though the electrode sections were deeply buried in a uniform expanse of formation material. Dimensions in the horizontal plane (x-y) were kept the same as for the top/bottom configuration (20 m x 20 m). However, to keep run times to a minimum, a smaller vertical (z) dimension was used. This was possible because the potential difference which generates the electric fields within the formation is between electrode regions on the well. This causes the electric fields to be more contained around the bottom 20 meter section of well than in the case of the top/bottom configuration. A suitable length for the vertical dimension in the multiple electrode simulations was found to be 40 meters. Here, the midpoint of the well section was located at z equal to 20 meters.

Shown in Figure 3.23(a) is the first multiple electrode configuration that was examined. It consisted of three 1 meter long electrodes separated by two insulated sections that were 8.5 meters in length.

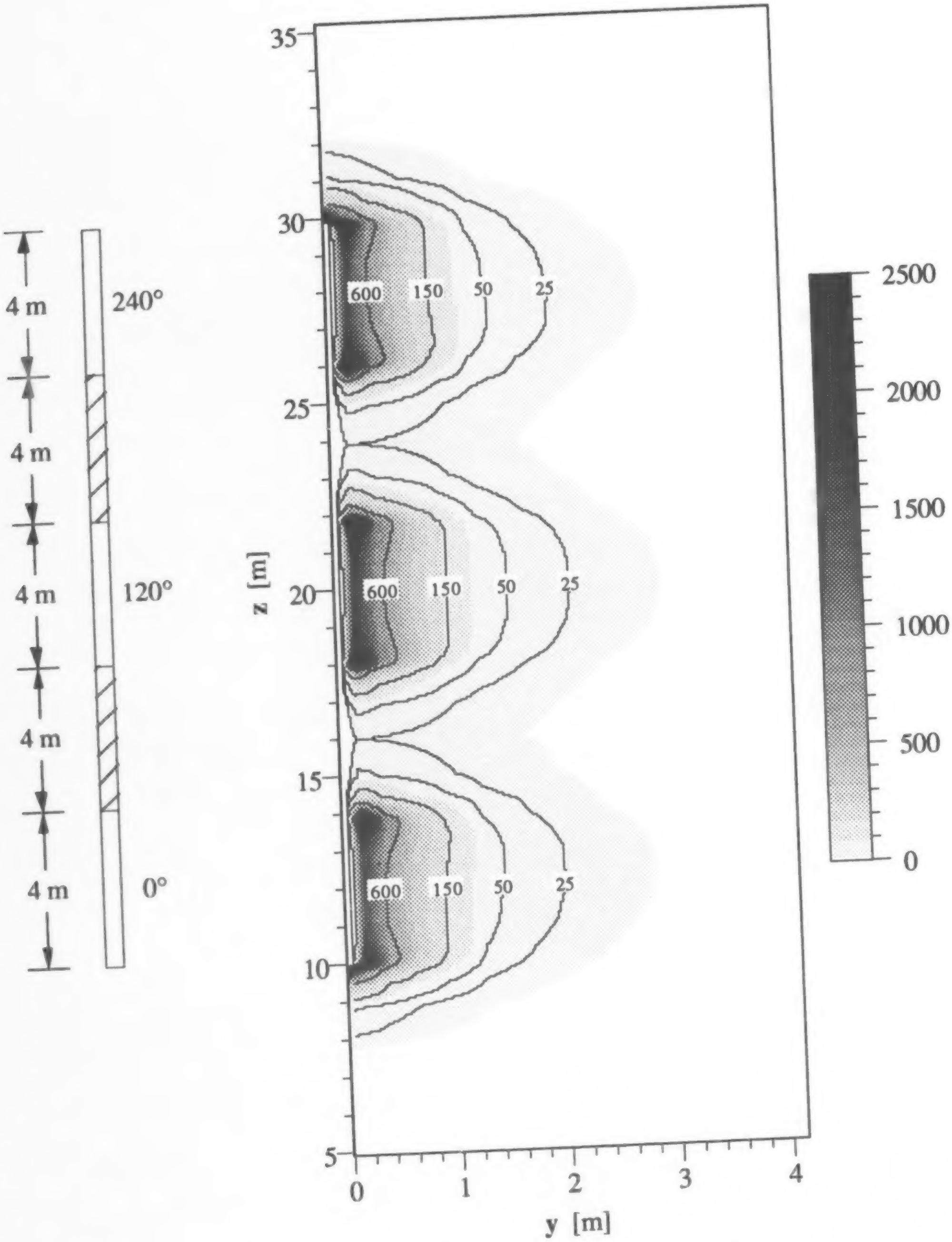
The heating pattern generated by three-phase excitation of this configuration is shown in Figure 3.23(b). In general it could be said that the heat distribution along the length of the well section is highly non-uniform. Heating is restricted to relatively narrow regions around each of the electrodes. In between these electrodes there is no significant heating. In addition, heating near to the electrodes is very intense. As a result, overheating of the electrodes leading to decoupling would be a major concern if using this configuration for sustained heating of underground formations.

Increasing the length of the electrodes to 4 meters as shown in Figure 3.24(a), improved the heating uniformity. From the greyscale plot in Figure 3.24(b), it can be seen that the heat distribution was such that there was heating along the entire length of the well section. As with previous multiple electrode configurations though, heating was still much more intense in the formation surrounding the electrode sections than in that surrounding the insulated sections.



(a) (b)

Figure 3.23 (a) Three Electrode Configuration for 20 m Well Section with Electrodes 1 m Long and Insulated Sections 8.5 m Long. Phase Angles with Respect to a Common Reference Are Shown for Three-Phase Excitation. (b) Heating Rates [W$\cdot$m^{-3}] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW.



(a)

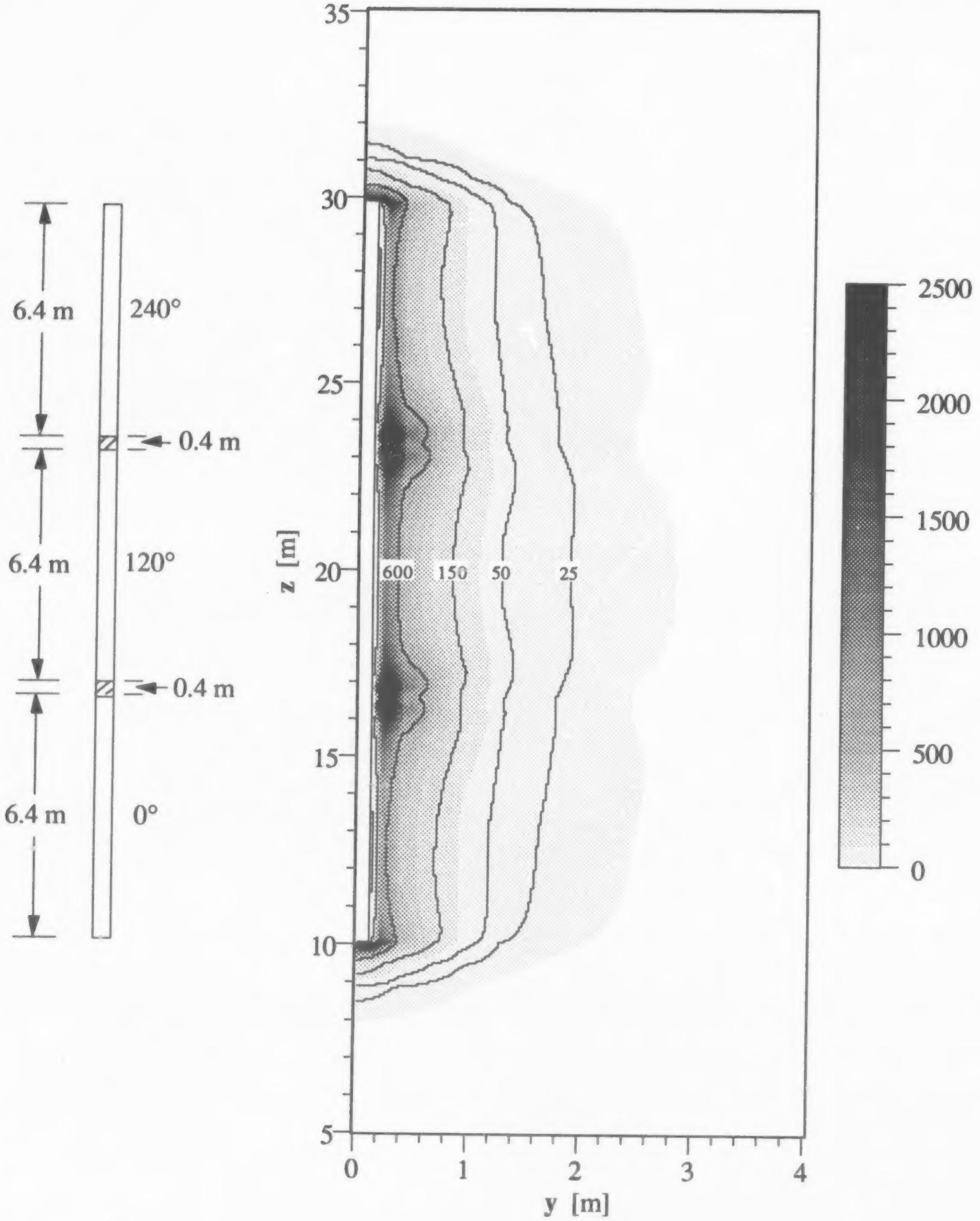
(b)

Figure 3.24 (a) Three Electrode Configuration for 20 m Well Section with Electrodes 4 m Long and Insulated Sections 4 m Long. Phase Angles with Respect to a Common Reference Are Shown for Three-Phase Excitation. (b) Heating Rates [W·m^{-3}] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW.

In the plot of heating rates, distinct hot spots can be seen at the ends of each electrode section. Like the top/bottom configuration, heating rates in the formation near the 4 meter electrodes were noticeably higher at the ends than at the middle. Once again, this occurrence would be a result of the higher current densities which exist at the ends of the finite length electrodes. It is interesting to note that for the top and bottom electrodes, heating is slightly more intense around the ends closest to the middle electrode than the other ends. This result is similar to that seen with the top/bottom configuration and it can again be attributed to the fact that current will tend to flow through the formation along the shortest (least resistive) path between electrodes.

In Figure 3.25(a) is shown a well section with electrodes that are 6.4 meters in length and insulated spacings that are 0.4 meters long. Three-phase excitation of this configuration produced the distribution of heating rates shown in Figure 3.25(b). With only 0.4 meters separating electrode sections on the well, heating rates in the regions between electrodes are higher for this configuration than for the previous configurations. This contributes to improved heating uniformity along the entire length of the well section. However, with the electrodes so close together, there is a greater tendency for the current to flow out the ends of the electrodes on either side of the insulated spacings. The closer the ends of the electrodes are, the less resistance to current flow there is between them. With more current flow out the ends of the electrodes, current densities and thus heating rates become higher in the formation near the well at these points and overheating can quickly occur.

Another aspect of the multiple electrode configuration for the single



(a)

(b)

Figure 3.25 (a) Three Electrode Configuration for 20 m Well Section with Electrodes 6.4 m Long and Insulated Sections 0.4 m Long. Phase Angles with Respect to a Common Reference Are Shown for Three-Phase Excitation. (b) Heating Rates [W·m^{-3}] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW.

well that was considered was load balancing. With multiple electrode sections on a single well bore, the load presented by the surrounding formation is not necessarily balanced. The reason for this is that the separation distance between all electrode pairs is not the same. For the three-electrode configuration studied here, the distance between the top and bottom electrodes is greater than the distance between the middle electrode and either the top or bottom electrode.

Evidence of load imbalance for the three-electrode configuration is given in Table 3.1. In this table, values of electrode impedance are given for the three different, three-electrode configurations for which heating rate distributions were previously determined. These values are calculated from electrode voltages and currents that the EPEIOS simulation program automatically determined for each configuration for the specified constant input power of 40 kW

As expected, given that the separation distance is larger, the impedance between the top and bottom electrodes was greater than between the other two electrode pairs for all three configurations. The load imbalance did, however, vary for the different configurations. When the electrodes were 1 meter long and the insulated spacings were 8.5 meters long, electrode impedances were almost the same and the load was essentially balanced. With an increased length in the electrodes and a decreased length of the spacings between them, the load became more unbalanced.

Although it appears that a balanced load can be achieved for a single well, multiple electrode configuration when electrodes are short in length relative to the insulated sections between them, the distribution of heating rates for such a configuration (refer to Figure 3.24(b)) is highly non-uniform thus making it impractical. On the other hand, the use of electrode sections that are not short relative to the insulated spacings provide more uniform heating but create an unbalanced load. It would seem that some compromise between heating uniformity along the length of the well section and load balancing has to be reached.

Configuration	Impedance Between Top and Bottom Electrodes [Ω]	Impedance Between Middle Electrode and Top or Bottom Electrode [Ω]
1 m Electrodes and 8.5 m Spacings	633	632
4 m Electrodes and 4 m Spacings	288	250
6.4 m Electrodes and 0.4 m Spacings	214	151

Table 3.1. Electrode Impedances in an Isotropic Medium for Various Three-Electrode Configurations with Three-Phase Excitation.

By manipulating the length of the electrode sections and the length of the insulated sections, a multiple electrode configuration was designed that generated fairly uniform heating and was approximately balanced in an isotropic problem domain when heating began. This configuration was arrived at by trial and error and no attempt was made to optimize the results.

As shown in Figure 3.26(a), top and bottom electrodes in the configuration are 5 meters long, insulated spacings are 3 meters long and the middle electrode is 4 meters long. Electrodes were excited with normal three-phase at a constant input power of 40 kW. The distribution of heating rates that is generated by this multiple electrode configuration in an isotropic medium is shown in Figure 3.26(b).

From the plot of heating rates it can be seen that heat is generated along the entire length of the 20 meter well section. The heating pattern appears somewhat non-uniform since heating rates are considerably higher around the electrode sections than around the insulated sections. However, when heating for extended periods of time, thermal conduction tends to reduce the temperature differential between regions where heating rates are high and regions where heating rates are lower. For the three-electrode configuration studied here, temperature gradients would be created between the formation surrounding the electrodes and the formation surrounding the insulated sections. These gradients would cause heat to be drawn away from the formation around the electrodes and into the regions surrounding the insulated sections. In this manner, the temperature rise along the length of the well section is made more uniform.

To further analyze the heating patterns generated by this three-electrode configuration when excited with three phases, heating for an extended period of time was simulated. As with the top/bottom

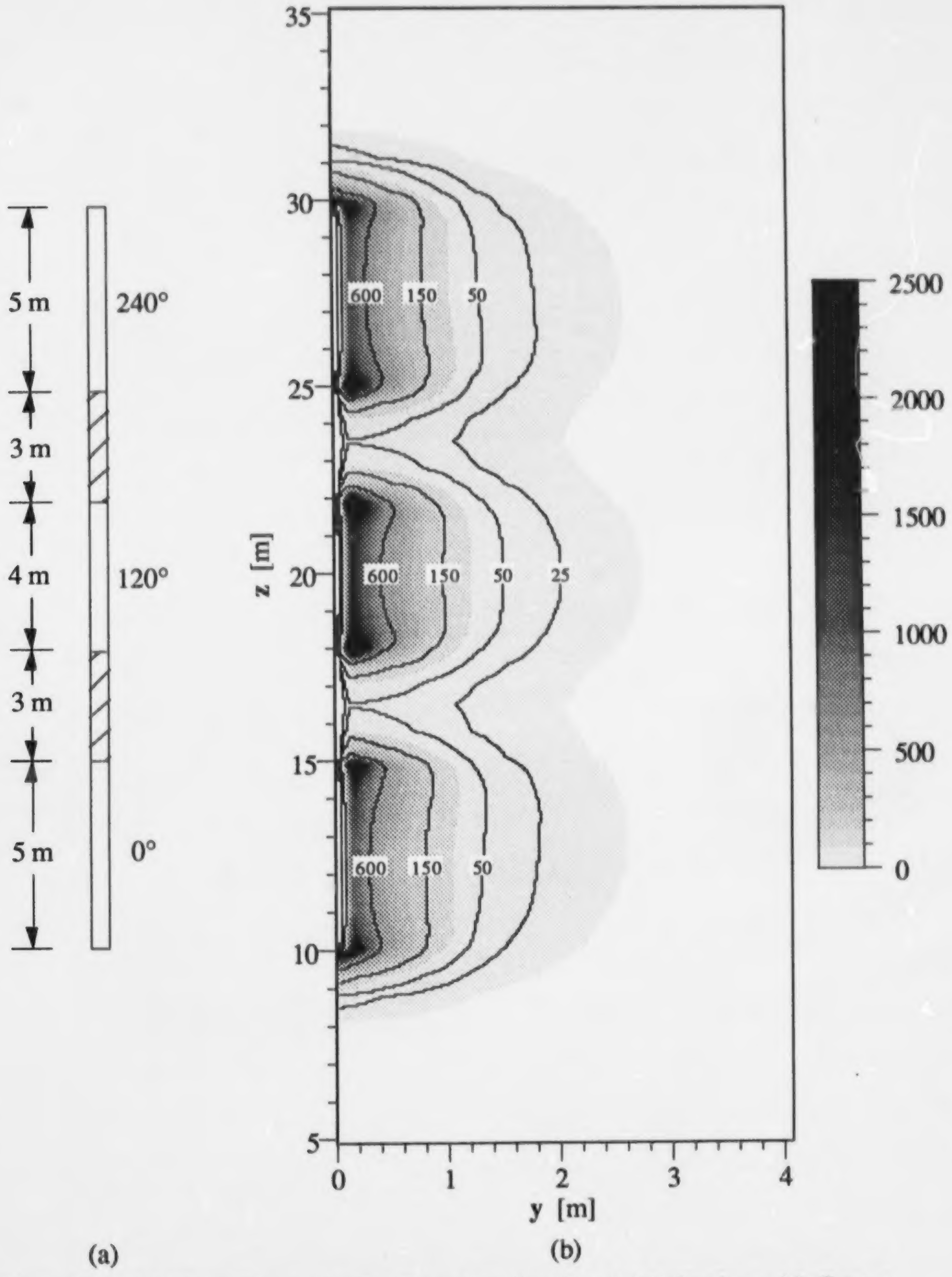


Figure 3.26 (a) Three Electrode Configuration for 20 m Well Section with Center Electrode 4 m Long, End Electrodes 5 m Long and Insulated Sections 3 m Long. Phase Angles with Respect to a Common Reference Are Shown for Three-Phase Excitation. (b) Heating Rates [W·m^{-3}] in Vertical (y-z) Plane for Well Section Shown in (a) in an Isotropic Medium. Total Input Power Was 40 kW.

configuration, an initial temperature of 10°C for the problem domain was assumed. The thermal and electrical characteristics of the formation material were the same as those used for the bottom layer in the simulations involving the top/bottom configuration. A constant input power of 40 kW was again used. The duration of the heating was 174.5 hours. This was the same amount of time as it took for the top/bottom configuration to generate a maximum formation temperature of 150°C.

The temperature distribution that resulted from the simulation is shown in Figure 3.27. As expected, the temperature distribution was more uniform along the length of the well section than the heating rate distribution would perhaps suggest. Temperatures were still higher around the electrode sections than around the insulated sections although significant temperature rise was achieved in the formation around the insulated sections as well.

Heating around the top and bottom electrodes was somewhat uneven as higher temperatures were generated around the ends of these electrodes that were closest to the middle electrode. This is a result similar to that seen with the top/bottom configuration where heating develops at one end of an electrode because of the fact that current tends to take the least resistive path between any two electrodes.

Significantly higher temperatures were generated around the middle electrode than around the other two electrodes. This is due to the shorter length of the middle electrode, which causes higher current densities and heating rates, as well as its position which is between the other two electrodes. The higher temperatures that were generated in this region caused the impedance between the middle electrode and the other two

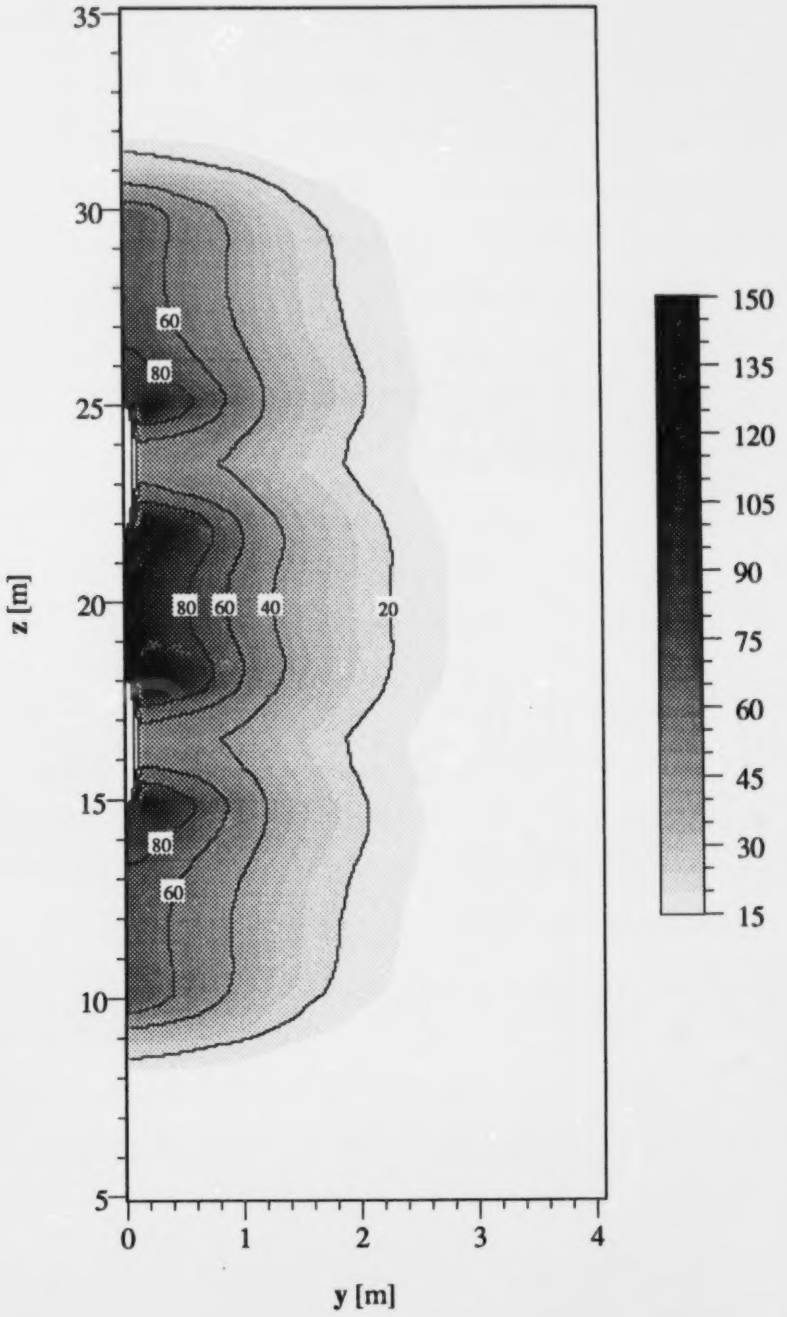


Figure 3.27 Temperatures [°C] in Vertical (y-z) Plane for Three-Electrode Configuration Shown in Figure 3.26(a) in an Isotropic Medium After Heating for 174.5 Hours. Power Input Was a Constant 40 kW.

electrodes to become much lower than the impedance between the top and bottom electrodes (77 Ω vs. 131 Ω). Thus, as heating progresses, the load becomes unbalanced. This is a problem that was encountered with all of the three-electrode configurations that were studied. No solution is readily apparent although further manipulation of electrode lengths and spacings may yield a configuration which maintains a balanced load throughout most of the heating process.

To achieve a direct comparison, a simulation was conducted with the three-electrode configuration in a layered problem domain that was the same as that used in simulations involving the top/bottom configuration. The 20 meter well section was positioned the same distance (2.5 m) below the interface between the two layers as before. It was assumed that the rest of the well bore was sufficiently insulated from the 20 meter electrode section. Electrodes were again driven with a constant 40 kW of three-phase power. The heating time was also the same at 174.5 hours.

The temperature distribution resulting from the simulation is shown in Figure 3.28. It is quite similar to the one generated by the three-electrode configuration in an isotropic medium (refer to Figure 3.27). Comparison of the two plots reveals that slightly higher temperatures are generated around the top electrode in the problem domain that was layered. The reason for this is that the relatively high conductivity of the overburden provides a lower resistance path for current leaving the top electrode. As a result, current flow through the top electrode and thus heating rates in the formation around it are higher than when the overburden was not included.

Inspection of the currents and voltages at the electrodes revealed that the presence of a high conductivity layer affects the electrode

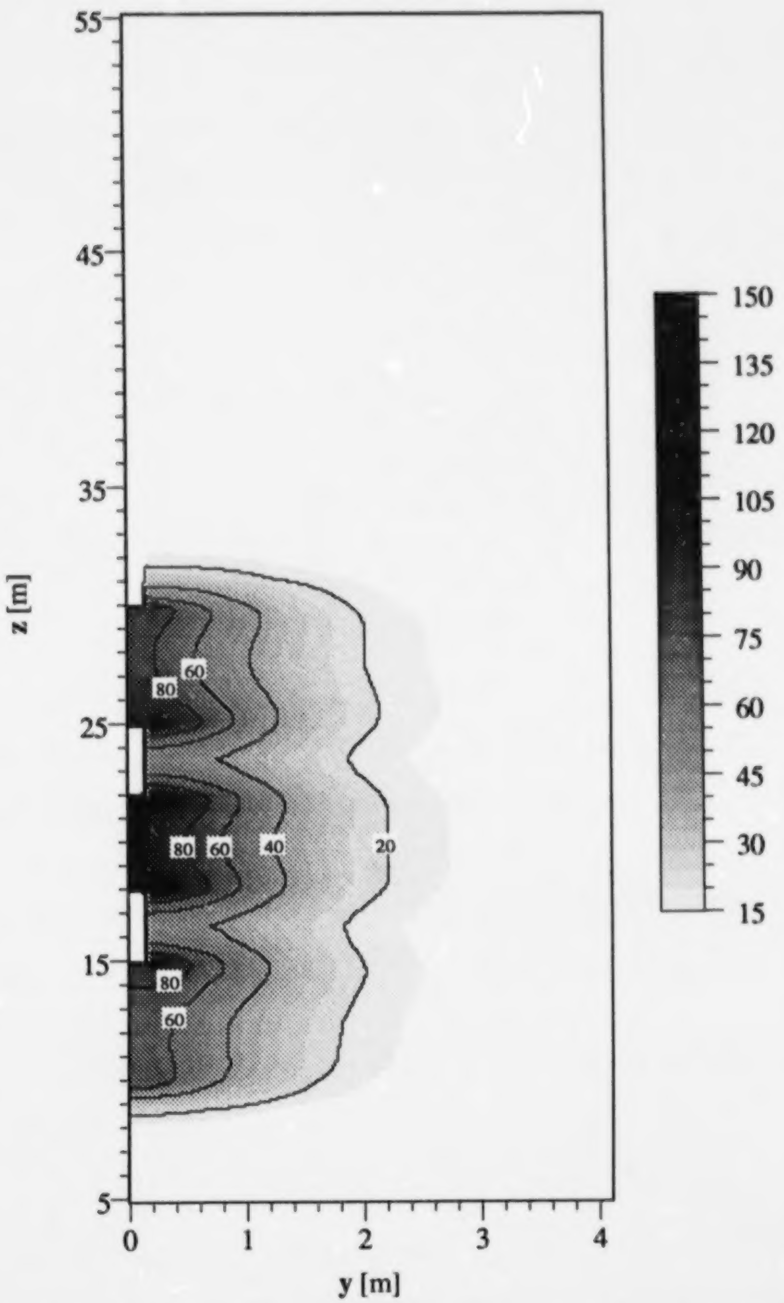


Figure 3.28 Temperatures [°C] in Vertical (y-z) Plane for Three-Electrode Configuration Shown in Figure 3.26(a) in a Layered Formation (Interface at z=32.5 m) After Heating for 174.5 Hours. Electrical Conductivity in the Top Layer Was Approximately Ten Times That in the Bottom Layer. Input Power Was a Constant 40 kW. Initial Temperature Was 10°C and Maximum Final Temperature Was 133°C.

impedances. Consequently, the load presented by this layered formation was not as balanced when heating was initiated as when the formation was entirely uniform. Once again, load balancing could probably be improved upon by adjusting electrode lengths and separation distances.

When compared to the top/bottom configuration (refer to Figure 3.22), it is apparent that the three-electrode configuration with three-phase excitation is better for heating along the entire length of the well section. With the top/bottom configuration, most of the heating is restricted to the top few meters of the electrode section. With the three-electrode configuration, heating similar to that seen only at the top few meters of the top/bottom configuration is generated around each of the three electrodes. Thus, the multiple electrode scheme produces useful heating along a greater percentage of the 20 meter well section. Heating along a longer section of well bore is desirable for oil recovery purpose since it would enable higher production rates.

In addition to better heating along the entire length of well section, the three-phase scheme also generated a maximum temperature at the end of the heating process that was lower than the top/bottom configuration (133°C vs. 150°C). The lower temperature is good since it will allow more heat to be delivered to the formation before overheating of the electrodes occurs.

Power Losses

Before making a definitive choice as to whether or not one form of electrode excitation is best suited for a specific application, consideration should be made of power losses in the delivery systems for the various electrode configurations. Generally speaking, such considerations would be very complex and beyond the scope of this thesis. However, by way of example, a study is made of the power losses that would exist in a typical delivery system for the three-electrode, three-phase configuration as compared to the top/bottom configuration. The study is not comprehensive as it is intended only to outline power loss considerations for two different configurations.

Power loss in a typical top/bottom configuration has been analyzed by Stroemich, Vermeulen, Chute and Sumbar (1990). They identified losses originating from electrical impedances in the production tubing (the path along which current is delivered to the electrode) and in the upper casing (the path along which current returns to the source). The electrical impedance in the tubing is simply the internal impedance of the tubing (Z_{tubing}). The impedance of the upper casing is not as obvious. The reason is that the current return path through the upper casing is very complex and details of this current flow are not well known. However, a reasonable first approximation for impedances in the upper casing was made by Stroemich *et al* (1990). This approximation consists of a series combination of the internal impedance of the upper casing (Z_{casing}) and the electrode impedance (R_{rod}) of the upper casing. Values for the impedances in a top/bottom configuration were determined by Stroemich *et al* (1990) for tubing and casing that is typically used in oil recovery. Assuming the use of the same tubing and casing, an electrode depth of 400 meters and an electrode current of 50 A rms, the following calculations were made for power loss in the top/bottom configuration that was studied here.

The production tubing used was a J-55 tubing with an outside diameter of 74 millimeters. For a current of 50 A rms, it has an impedance of approximately $4.6 \times 10^{-4} + j4.1 \times 10^{-4}\ \Omega\cdot\mathrm{m}^{-1}$ (Stroemich *et al*, 1990). Thus for an electrode section that is 400 meters below the surface, the total impedance of the tubing would be:

$$\begin{aligned} Z_{tubing} &\approx 400(4.6 \times 10^{-4} + j4.1 \times 10^{-4})\ \Omega \\ &\approx 0.18 + j0.16\ \Omega \end{aligned}$$

The upper casing used was an L-80 casing with an outside diameter of 179 millimeters. For a current of 50 A rms, this casing has an internal impedance of approximately $1.0 \times 10^{-4} + j1.2 \times 10^{-4}\ \Omega\cdot\mathrm{m}^{-1}$. Since current is collected on the outside surface of the casing and then returns to the source along the inside surface of the casing, the length of the casing is approximately doubled (Stroemich *et al*, 1990). The total internal impedance of the casing would then be:

$$\begin{aligned} Z_{casing} &\approx 2(400)(1.0 \times 10^{-4} + j1.2 \times 10^{-4})\ \Omega \\ &\approx 0.08 + j0.10\ \Omega \end{aligned}$$

A value for the electrode resistance associated with the upper casing is calculated from the following expression for electrode resistance,

$$R_{rod} \approx \frac{\ln\dfrac{2l}{a} - 1}{2\pi\sigma_{form}\, l} \tag{3.1}$$

where l is the length of the casing [m], a is the radius of the casing [m] and σ_{form} is the conductivity of the formation material [$S{\cdot}m^{-1}$]. This equation is valid for $l \gg a$ and although it applies to the case where the medium surrounding the electrode is somewhat thicker than the length of the electrode, it used here to obtain a good approximation. For an overburden with an electrical conductivity equal to 1.0×10^{-2} $S{\cdot}m^{-1}$, the electrode resistance of the casing may then be calculated as:

$$R_{rod} \approx 0.32\ \Omega$$

The total power losses for the top/bottom configuration ($P_{top/bottom}$) would then be:

$$\begin{aligned} P_{top/bottom} &\approx I^2(R_{tubing} + R_{casing} + R_{rod}) \\ &\approx 50^2(0.18 + 0.08 + 0.08) \\ &\approx 1450\ \text{W} \end{aligned} \tag{3.2}$$

The efficiency of power delivery for the top/bottom configuration ($\eta_{top/bottom}$) may be calculated as:

$$\eta_{top/bottom} \approx \frac{R_{electrode}}{R_{electrode} + R_{tubing} + R_{casing} + R_{rod}} \tag{3.3}$$

where $R_{electrode}$ is the electrode resistance [Ω]. From the simulation of the top/bottom configuration in a layered medium, the electrode resistance was found to be approximately 31 Ω at the start of heating. Using this value, the power delivery efficiency is calculated to be:

$$\eta_{top/bottom} \approx 98.2\%$$

Power delivery for the three-electrode configuration is somewhat different from that used for the top/bottom configuration. As a result, sources of power loss are different as well. These sources of power loss may be identified by consideration of current flow in the three-electrode configuration.

With the three-electrode arrangement, electrical power would be delivered to the electrodes via a three-phase cable. Such a cable would have some inherent impedance and consequently some power loss would occur in the cable. Current that is delivered to the formation through the electrodes returns through the electrodes as well. Thus there would be no power loss along the current return path. Although the cable would be insulated, current flow through the cable would induce some eddy currents in the production tubing and the well casing. These eddy currents and the losses associated with them are the subject of a current study in the Applied Electromagnetics Laboratory at the University of Alberta. However, little is known about them at this time and so power loss resulting from the creation of eddy currents is not considered here. No other significant sources of power loss exist for this configuration. As a result, power losses in the three-electrode configuration are assumed to consist solely of losses in the power delivery cable.

Before calculating the losses in the cable, a suitable cable for this application had to be chosen. This was done by consideration of the electrical and physical requirements for the cable. The electrical requirements were based on an assumed maximum input power of 100 kW and an electrode impedance that may range between 50 and 250 ohms. The chosen cable was an RW90, single core cable with a 6 AWG copper core and thermoset (XLPE) insulation. The voltage rating for this cable is 5 kV and the maximum current rating is 65 A at 30°C (*CE code handbook,* 1990). The XLPE insulation was chosen because of its ability to withstand higher temperatures (it is good to 90°C). Single core cables were used as opposed to a three core bundle to ensure that the cable would be able to easily fit in the space between the chosen well casing and production tubing.

The losses in this cable may be simply approximated from the dc resistance of the copper cores. Proximity effect and the skin effect are typically neglected for cables with a cross sectional area that is less than 150 mm^2. Consequently these effects are not considered here. Losses in the sheath surrounding each cable may also be ignored because of the relatively low voltage that is used (King & Halfter, 1982).

To compare the power losses between the two different configurations, the same power was delivered to the formation in both cases. With an electrode current of 50 A rms and an electrode resistance of 31 Ω, the power delivered by the top/bottom configuration was 77.5 kW. Using the power delivered to the formation and the electrode resistance for the delta-connected, three-electrode configuration, the line current may be calculated as:

$$I_L = \sqrt{\frac{P_{in}}{R_{phase}}} \tag{3.4}$$

where I_L is the line current [A], P_{in} is the total power delivered to the formation [W] and R_{phase} is the electrode resistance for each phase [Ω]. Although a balanced three-electrode configuration was not obtained for the layered medium, it was assumed as a first approximation that the electrode impedance in the layered medium would be equal to that in the uniform medium. It was felt that this was reasonable since the electrode impedances for the configuration that was approximately balanced in the uniform medium did not change much in the layered medium. The reason for this would be the fact that with the three-electrode arrangement, most of the current flow is restricted to the region near to the electrodes. Using this assumption, the electrode impedance for each phase of the three-electrode configuration was approximately 230 Ω. Thus, the line current may be determined as being:

$$I_L \approx 18 \text{ A rms}$$

The dc resistance of each of the three copper cores is 1.35 $\Omega \cdot \text{km}^{-1}$ (Bungay & McAllister, 1990). For an electrode section that is 400 meters beneath the surface, the total resistance of each core (R_{cable}) would be:

$$\begin{aligned} R_{cable} &\approx 0.4(1.35)\ \Omega \\ &\approx 0.54\ \Omega \end{aligned}$$

The total power lost in the three cables ($P_{3\emptyset}$) would then be:

$$\begin{aligned} P_{3\emptyset} &\approx 3(I_L{}^2 R_{cable}) \\ &\approx 525\ \text{W} \end{aligned} \tag{3.5}$$

Similar to the top/bottom configuration, the efficiency of power delivery for the three-electrode configuration ($\eta_{3\phi}$) may be calculated to be:

$$\eta_{3\phi} \approx 99.7\ \%$$

It is apparent that there is greater power loss with the top/bottom configuration than with the three-electrode configuration even though the cable resistance is almost equal to the sum of the resistances which cause power loss in the top/bottom configuration. This is mostly due to the fact that the electrode resistance for each phase of the three-electrode arrangement is much higher than for the top/bottom configuration (230 Ω vs. 31Ω). As a result of the higher electrode resistances, less current is required to deliver the same amount of power to the formation. Since power losses are proportional to the square of the current, a lower current causes a significant reduction in power loss.

It is conceivable that power loss in the three-electrode configuration could be reduced by using a power delivery cable with a higher current rating. Such a cable would have a larger cross sectional area and consequently lower losses. The choice of cable would however, depend on economic considerations as well.

CHAPTER 4

Interpretation of the Results

Summary and Conclusions

Given that conventional oil recovery techniques suffer from serious limitations when used as a means of recovering oil from deeply buried oil sand formations, research has been conducted into developing alternative oil recovery methods. One such method that has been found to be potentially viable uses low frequency electromagnetic heating as a preheat to a steam or hot water drive. In the past, single-phase excitation of embedded electrodes has been proposed as the means to deliver electrical energy to underground formations. However, for large volumes of oil sand *in situ,* it has been found that it is difficult to obtain an optimal distribution of heat using single-phase excitation.

It was thought that use of multiphase instead of single-phase excitation to drive electrodes would enable improved heating of underground formations. To test this theory, several electrode arrays and phasing schemes were studied. In addition to array patterns for heating large volumes of oil sand, multiphase excitation of circular arrays and an array of electrodes along a single well were also examined.

To carry out the investigation, computer simulation was used. The simulator numerically solved Maxwell's equations and the heat equation for a specified problem domain using the finite difference approximation technique.

The first group of arrays that were examined were those that could be used in the heating of large volumes of oil sand *in situ.* They corresponded with vertical well placement patterns that are commonly used in conventional oil recovery schemes. Both single-phase and multiphase excitation of these arrays was simulated.

From the simulations it was discovered that for a given array configuration, more current flow pathways can be generated through the formation with multiphase excitation than with single-phase excitation. This can cause the multiphase excitation scheme to produce a more uniform distribution of heat within the array. In addition, the multiphase excitation tends to distribute the current more evenly between the electrodes in the array. This contributes to lower heating rates at the electrodes when multiphase excitation is used than when single-phase excitation in used. The results of the simulations also revealed that current flow and thus, heating within an array excited by multiple phases can be tailored by adjusting the phase relationship between electrodes. Consequently, more uniform heating can generally be achieved using multiphase excitation than single-phase excitation.

The next group of arrays that were studied were circular arrays involving a varying number of electrodes. These were excited such that each electrode in the array was at a unique phase angle with opposing electrodes being 180° out of phase. Simulation results revealed that this type of configuration was capable of generating a zone of uniformly intense heating within the confines of the array. This result was observed because current flow is generated between all electrode pairs in the circular array and current flow along any one particular path does not dominate over current flow along another path. Simulations involving the circular array also revealed that for the phasing scheme used, heating was more effective when a greater number of electrodes (and phases) were employed. This is evidenced by the fact that for the twelve electrode configuration, average heating rates in the bulk of the heated material were higher and heating intensity around the electrodes lower than for the other configurations which employed fewer electrodes.

The last array type examined was the single well. Here, three electrode regions were defined on a well section and excitation of the electrodes by standard three-phase was simulated.

Initial simulations involving the three-electrode configuration in a uniform medium revealed that heating generated along the length of the well section is not very uniform when electrodes are short relative to the length of the insulated spacings. The use of longer electrodes improves heating uniformity although it tends to make the three-phase load presented by the surrounding formation less balanced. It was found that by manipulating the length of electrodes and spacings, reasonably uniform heating and a balanced load can be achieved. Load balancing does, however, depend somewhat on the composition of the formation. Heated volumes near a region of substantially different electrical conductivity will have slightly different electrode resistances than heated volumes that are seemingly infinitely thick.

When compared to the top/bottom configuration, it was found that three-phase excitation of a three-electrode configuration can produce a better temperature distribution along the length of well over which heating is desired. With three electrodes and three-phase excitation, significant temperature rise is achieved over a greater percentage of the well section and overheating of the electrodes does not occur as quickly. In addition, it was discovered that electrical power can be delivered to the formation more efficiently with the three-electrode configuration than with a top/bottom configuration. This is due largely to the fact that electrode impedances are much higher for the three-electrode arrangement.

Recommendations

This investigative study has revealed many features of multiphase excitation as it pertains to low frequency electromagnetic heating. There is, however, more work that can be done.

For completeness, the results obtained here by computer simulation should be verified. This could probably best be done by physical scale modelling of the various configurations. Equations for physical modelling of the low frequency, electromagnetic heating process have been defined by Vermeulen *et al.* (1979).

More comprehensive simulations involving multiphase excitation of electrode arrays should be conducted. These simulations would involve extended periods of heating and would include consideration of heat flow by thermal conduction through the heated material. Those configurations that are intended for use in aiding oil recovery should also be examined with a simulator that takes into account fluid flow.

Further analysis of circular arrays excited with multiple phases should be performed. The highly uniform heating that can be achieved within the array may be useful for heating materials other than oil sand. Potential areas of application may include the thermal treatment of cancerous tissue and heat treatment of contaminated soils.

In the case of the multiple electrode configuration for the single well, further study should be conducted on load balancing in conjunction with heating uniformity along the length of the well section. This would include an examination and possible optimization of electrode length and placement along the well section over which heating is desired.

Simulation of heating using electrode arrays and multiphase excitation schemes other than those used in this study should also be carried out. There are many possible combinations. These should be explored to determine optimal electrode placement and phase assignment for obtaining specific patterns of heat distribution.

References

AOSTRA: a 15 year portfolio of achievement. (1990). N.p.: n.p.

Bridges, J. E., Sresty, G. C., Spencer, H. L., & Wattenbarger, R. A. (1985). Electromagnetic stimulation of heavy-oil wells. In R. F. Meyer (Ed.), *The Third UNITAR/UNDP International Conference on Heavy Crude and Tar Sands, July 1985, Long Beach California,* 615-622. (AOSTRA Library and Information Services No. 012577)

Bungay, E. W. G. & McAllister, D. (Eds.). (1990). *Electric cables handbook* (2nd ed.). Boston: BSP Professional Books.

Carrigy, M. A. (1983). Thermal recovery form tar sands. *Journal of Petroleum Technology, 35,* 2149-2157.

CE code handbook an explanation of the rules of the CE code, part I. (1990). Rexdale, ON: Canadian Standards Association.

Chute, F. S., & Vermeulen, F. E. (1988). Present and potential applications of electromagnetic heating in the *in-situ* recovery of oil. *AOSTRA Journal of Research, 4(1),* 19-33.

Chute, F. S., Vermeulen, F. E., & Stevens, L.G. (1987). A study of the technical and economical feasibility of an electric preheat process for *in situ* recovery from Athabasca oil sands. *AOSTRA Journal of Research, 3(3),* 139-154.

Chute, F. S., Vermeulen, F. E., McPherson, R. G., Hiebert, A. D., & Fearn, J. (1985). *Electromagnetic heating of oil sands.* (AOSTRA Research Agreement No. 296). Applied Electromagnetics Laboratory, Department of Electrical Engineering, University of Alberta.

Gill, H. (1983). The electrothermic system for enhancing oil recovery.

Journal of Microwave Power, 18(1), 107-110.

Glandt, C. A. & Hsu, C. F. (1992). Electric preheating in low-injectivity tar sand deposits. *The SPE/DOE Eighth Symposium on Enhanced Oil Recovery.* April 1992, Tulsa, Oklahoma.

Hiebert, A. D. (1986). Modelling of the electric pre-heat process. Doctoral dissertation, University of Alberta.

King, S. Y. & Halfter, N. A. (1982). *Underground power cables.* New York: Longman Inc.

Khosla, A., & Towson, D. E. (1985). Application of computer modelling to the PCEJ electric preheat/steamdrive Athabasca pilot. *The 36th Annual Technical Meeting of the Petroleum Society of CIM.* June 1985, Edmonton, Alberta.

McPherson, R. G., Chute, F. S., & Vermeulen, F. E. (1986). The electromagnetic flooding process for in-situ recovery of oil from Athabasca oil sand. *Journal of Microwave Power, 21,* 129-147.

Pritchett, W. C. (1976). *Method and apparatus for producing fluid by varying current flow through subterranean source formation.* United States Patent No. 3948319, Issued April 6, 1976.

Settari, A., & Karcher, B. (1985). Simulation of enhanced recovery projects - the problems and the pitfalls of the current solutions. *The Journal of Canadian Petroleum Technology, 37,* 22-28.

Spencer, H. L., Bennett, K. A., & Bridges, J. E. (1988). Application of electromagnetic stimulation to Canadian heavy oil reservoirs. *The Fifth Annual Heavy Oil & Oil Sands Technical Symposium, March 16, 1988.* (AOSTRA Library and Information Services No. 012016)

Stroemich, C. P., Sumbar, E., Vermeulen, F. E., & Chute, F. S. (in press).

High pressure high temperature cell for electrical conductivity measurements of earth-type materials. *IEEE Transactions on Geoscience and Remote Sensing*.

Stroemich, C. P., Vermeulen, F. E., Chute, F. S., Sumbar, E. (1990). Wellbore power transmission for *in-situ* electrical heating. *AOSTRA Journal of Research, 6(4)*, 273-294.

Strom, N. A., & Dunbar, R. B. (1979). Bitumen resources of Alberta: converting resources to reserves. In *The Future of Heavy Crude and Tar Sands* (pp. 47-60). New York: McGraw Hill Inc.

Vermeulen, F. E., & Chute, F. S. (1983). Electromagnetic techniques in the in-situ recovery of heavy oils. *Journal of Microwave Power, 18(1)*, 15-29.

Vermeulen, F. E., Chute, F. S., & Cervenan, M. R. (1979). Physical modelling of the electromagnetic heating of oil sand and other earth-type and biological materials. *Canadian Electrical Engineering Journal, 4(4)*, 19-28.

Yasuda, S. (1978). *Method and equipment to extract and collect petroleum substance from underground tar sand stratums and underground petroleum reservoirs*. Canadian Patent No. 1034870, Issued July 18, 1978. (Alberta Oil Sands Information Centre No. 006154)

APPENDIX A

Changes Made to EPEIOS in Transferring It From the Amdahl to the Macintosh Computer

Version 1.0

This was the first version to run on the Macintosh system. The original EPEIOS program was developed on the University of Alberta's mainframe computer (Amdahl). The FORTRAN code was compiled on the Amdahl using the VS FORTRAN compiler (an IBM product) under the MTS operating system. The Language Systems (LS) FORTRAN 2.0 compiler running under the MPW environment was chosen to compile EPEIOS on the Macintosh since it supported namelist input and provided several useful extensions to standard FORTRAN. As a result of the differences between the two compilers and operating systems, several changes had to be made to the EPEIOS code before the program would compile on the Macintosh. The following is a brief outline of the changes that were made.

- common blocks were inserted in the subroutines as required by using an INCLUDE statement. This replaced the previous method of compiler directives (C/ INSERT ...) used in conjuction with the Olympic Pre-processor (OPP).

- in subroutine INHEAT, the default value of the variable DTIMAX, which represents the maximum size of the timestep, was set to 1.0E35 instead of 1.0E40 because 1.0E40 exceeded the limit for a REAL*4 variable.

- in subroutines INHEAT and SCALES, the intrinsic function 'IMAG' which selects the imaginary part of a complex number was not recognized by the LS compiler so it was replaced by 'AIMAG' which performs the same function.

- in subroutines RUNTIM, INPUT, TEST and OUTPUT, all lines of the code relating to the cost of the run (in MTS soft dollars) were deleted or commented out.

- in subroutine INPUT, code that enabled EPEIOS to be run without terminal interaction (batch run), was commented out since such a feature would be of little use on a Macintosh system. In doing this, a call to an MTS subroutine (CREPLY) was removed.

- in subroutine INPUT, logical I/O units were assigned using the OPEN command. The program will write to EPEIOS:EPEIOS.OUT: /OUTPUT for printed output and to EPEIOS:EPEIOS.OUT:STORE if data storage output is requested. The program will read a restart record from EPEIOS:EPEIOS.OUT:RESTART and the input file is chosen via a standard dialog box at runtime.

- in subroutine OSTOR, format statements were added for each write statement to the data storage unit IOSTOR. Correspondingly, format statements were added to subroutine INREC for read statements from the data storage unit IOREST.

- the variable LERROR was deleted from common block COMSYS as it did not appear to be used anywhere in the program.

- in subroutine RUNTIM, new code was written to determine the running time of the program since original code made use of an MTS subroutine (TIME). The replacement code utilizes a subroutine that is predefined in the LS FORTRAN compiler (SECNDS). In addition, a new variable was introduced (BEGIN) and added to the common block COMSYS. Due to the longer running times, CPU time is displayed in minutes instead of seconds.

- in subroutine DAYTIM, calls to MTS subroutine TIME were replaced with calls to LS FORTRAN compiler subroutines DATE and TIME.

- in subroutine OUTPUT, some adjustments had to be made to the format statements so that for the chosen font (MONACO), the printed output would appear as it did when it was generated on the MTS system.

- in order to reduce the compilation time of EPEIOS after changes have been made to the common blocks or the program itself, the code was broken up so that each subroutine or common block occupied a separate file bearing the name of the subroutine or common block with an attached ending of '.F'. All subroutine files were placed in the folder EPEIOS:SUBS and all common block files were placed in the folder EPEIOS:COMMONS. An MPW make file was created (EPEIOS.make) which combines and compiles all the files on execution of the BUILD command. Compilation time is reduced when using a make file because only the files which have been altered since the time of last compilation are recompiled. The common blocks are listed as dependencies, so changes to the common blocks will be seen as a change to the subroutine (file) in which they are included.

- to facilitate ease of operation, several MPW scripts were created to support the main program. All scripts are located in the folder MPW:EPEIOS.Scripts. Note that this folder must be included in the Set Commands line of the MPW file 'Startup' before they can be executed. The following is a list of the scripts and their functions:

  Clearout - empties the output file (EPEIOS.OUT: /OUTPUT) so that all old data can be removed before new data is written in.

  Newcom - opens and displays the script Initcom so that the number of grid blocks in the x, y and z directions as well as the number of regions required may be specified to a new value.

  Initcom - prepares the script Param to insert new values for number of grid blocks in the x, y and z directions and for number of regions, into the common blocks. The script Initcom achieves this by replacing the file Param with a copy of the file Param.o which has grid block and region parameters represented by the variables xx, yy, zz and ww. When Initcom is executed, these variables are replaced with the values that are specified in the script. It then executes Param.

  Param - this script replaces the currently used common block files (ending in .F) with a copy of corresponding common block files (ending in .O) that have the number of grid blocks in the x, y and z directions and the number of regions represented by the variables qx, qy, qz and qr respectively. When executed, the script replaces these variables with values corresponding to xx, yy, zz and ww as specified in the script Initcom.

Note that for changing array dimensions in the common blocks, only the scripts Newcom and Initcom need to be used. The script Param is carried out automatically by Initcom.

- in subroutine INPUT, variable VERSN was set equal to 1.0.

APPENDIX B

Developmental Changes Made to the EPEIOS Code

Version 1.1

Often times when complex structures (non-rectangular) are simulated, many different regions need to be defined. In such instances, the region data which is printed to the output file becomes extensive as well as being somewhat repetitive, especially when repeated runs are performed for the same grid structure. For cases like these as well as others where the complete region data is not desired in the output record, a provision was made whereby the user may suppress the printing of region data in the output. To achieve this, a new variable had to be added and changes had to be made to subroutine INPUT.

The new variable is called LPREG and is located in common block COMSYS. It is a logical variable whose value the user may specify in the input namelist &INPUT2. If LPREG is set equal to .FALSE., no region data will be printed in the output record. If LPREG is set equal to .TRUE., all region data is printed out. The default value is LPREG=.TRUE..

In addition to the introduction of a new variable into the code, the variables which represented the coefficients used in the equation to describe the temperature dependence of the electrical conductivity were changed. The change was performed for the sake of clarity and to emphasize the difference between the coefficients now used in the cubic equation and those used originally in the MTS version.

The following is an outline of the changes that were made to EPEIOS.

- the variable LPREG was added to common block COMSYS in file COMSYS.F

- in subroutine INPUT, the variable LPREG was added to the namelist INPUT2.

- in subroutine INPUT, the variable LPREG was given a default value of .TRUE..

- in subroutine INPUT, the value of LPREG was checked prior to any calls to subroutines that printed region data in the output record.

- in subroutine INPUT, the variables ALPHA1, ALPHA2 and ALPHA3 were changed to ALPHAB, ALPHAC and ALPHAD respectively.

- in subroutine INPUT, variable VERSN was set equal to 1.1.

Version 1.2

The EPEIOS program uses the finite difference method to calculate a value for electric potential at the centre of each grid block in the user defined grid structure. From the electric potential, heating rate and temperature at the centre of the grid blocks are calculated as well. If a value was desired to be known for potential, heating rate or temperature at coordinates that did not correspond to the centre of a grid block, a value was determined for that point by performing a trilinear interpolation. In the original interpolation subroutine, the centres of the eight grid blocks that were immediately surrounding the point at which a value was desired were used as interpolation points. This method worked well except when the point to be interpolated was located near a boundary between an insulated region and a conducting region. Since electrically insulated blocks are assigned a value of zero for potential and heating rate and thermally insulated blocks are assigned a temperature of zero, interpolation using an insulated block and a conducting block would yield a value that is somewhere between the value calculated at the centre of the conducting block and zero. As a result, the required condition that potential and temperature not change across an insulated boundary did not appear to be met in the region between the centre point of an insulated block and the centre point of a conducting block even though the calculated values at the centres of the grid blocks were correct. To eliminate this inconsistency, a more sophisticated interpolation subroutine was needed.

The new trilinear interpolation subroutine examines the properties of the grid block in which the point of interpolation is located, as well as the properties of the surrounding grid blocks. If the point of interpolation is located within the bounds of an insulated grid block, it will automatically be assigned a value of zero. If the point is located within the bounds of a conducting grid block but one or more of the blocks surrounding it are insulating, the program systematically tries different sets of interpolation points in an effort to perform the interpolation with grid blocks that are all conducting. Interpolations at points where all surrounding grid blocks are conducting are carried out as before. Testing of this interpolation subroutine showed that the necessary boundary condition between insulating and conducting grid blocks was more accurately maintained than when the original interpolation subroutine was used. An outline of the changes that were made are as follows:

- the original subroutine LININT was completely replaced with a new trilinear interpolation subroutine. Refer to the program listing for definitions of the new variables that were introduced as well as a complete description of the methodology used in carrying out the interpolation.

- an array variable describing the electrical or thermal properties of the grid blocks was added to the list of input/output parameters in all calls to subroutine LININT

- two new subroutines were created (CHECK and PICK) to support the main interpolation routine LININT. Subroutine CHECK examines the electrical or thermal properties of a set of eight grid points. Subroutine PICK establishes the array coordinates of the values to be used the interpolation based on the results produced from CHECK. All three subroutines are located in the file LININT.F.

- in subroutine INPUT, variable VERSN was set equal to 1.2.

```
      SUBROUTINE LININT(A,A2,A3,XT,YT,ZT,NPX,NPY,NPZ)
C
C     This subroutine does a linear interpolation in 3D.
C     Description of parameters:
C
C  A           - Array to be interpolated.
C  A2          - Array describing electrical or thermal properties of grid blocks
C  A3          - Result of interpolation.
C  XT,YT,ZT    - Coordinates of interpolation points.
C  NPX,NPY,NPZ - Number of points in each direction at which values are desired.
C
      INCLUDE ':COMMONS:COMGEO.F'
      INCLUDE ':COMMONS:COMSYS.F'
C
C
      REAL A3(NDX,NDY,NDZ),A(NDX,NDY,NDZ)
      REAL XT(NDX),YT(NDY),ZT(NDZ)
      INTEGER B(27),FLAG,H(7),PX,PY,PZ,A2(NDX,NDY,NDZ)
C
C  Perform interpolation for each point in succession.
C
       DO 1 I=1,NPX
          DO 1 J=1,NPY
             DO 1 K=1,NPZ
C
C  Find the first grid point at which the x,y,z coordinates are greater than
C  those of the interpolation point.
C
             DO 2 I1=2,NX
                IN = I1
                IF (XT(I).LT.XCOORD(I1))  GOTO 3
 2           CONTINUE
C
 3           DO 4 J1=2,NY
                 JN = J1
                 IF (YT(J).LT.YCOORD(J1))  GOTO 5
 4           CONTINUE
C
 5           DO 6 K1=2,NZ
```

```
                KN = K1
                IF (ZT(K).LT.ZCOORD(K1)) GOTO 7
  6         CONTINUE
C
C Determine the array coordinates of the grid block in which the interpolation
C point is located (PX,PY,PZ).
C
  7         IF ((XCOORD(IN)-XT(I)).GT.DELTAX(IN)/2) THEN
               PX = IN - 1
            ELSE
               PX = IN
            END IF
            IF ((YCOORD(JN)-YT(J)).GT.DELTAY(JN)/2) THEN
               PY = JN - 1
            ELSE
               PY = JN
            END IF
            IF ((ZCOORD(KN)-ZT(K)).GT.DELTAZ(KN)/2) THEN
               PZ = KN - 1
            ELSE
               PZ = KN
            END IF
C
C Check the electrical or thermal properties of the grid block in which the
C interpolation point is located.  If the value of A2 at array coordinate
C PX,PY,PZ is zero, then the interpolation point is located in an electrically
C or thermally insulated region.  If this is the case, the interpolation
C point is assigned a value of zero and the program advances to the next
C interpolation point.  If this is not the case, then the point must be
C located in a grid block that is conducting.
C
            IF (A2(PX,PY,PZ).EQ.0) THEN
                A3(I,J,K) = 0
                GOTO 1
            END IF
C
C Since IN,JN,KN are used to represent the outermost array coordinate of the
C points used for the interpolation, IN,JN,KN must all have a value of at
C least 2.
```

```
C
            IF (IN.EQ.1) IN = 2
            IF (JN.EQ.1) JN = 2
            IF (KN.EQ.1) KN = 2
            IM = IN - 1
            JM = JN - 1
            KM = KN - 1
C
C  Examine the electrical or thermal properties of all the eight grid blocks
C  immediately surrounding the interpolation point.  If none of these grid
C  blocks are designated as insulating, the interpolation may be carried
C  out.
C
            DO 10 L=IM,IN
              DO 10 M=JM,JN
                 DO 10 N=KM,KN
                 IF (A2(L,M,N).EQ.0) GOTO 13
 10         CONTINUE
            GOTO 11
C
C  If one or more of the 8 grid blocks that were to be used in the original
C  interpolation are found to be insulating, neighbouring grid blocks are
C  systematically examined to determine if they are conducting and thus
C  suitable for use in performing the interpolation.
C
 13         IF ((PX.EQ.IN).AND.(PY.EQ.JN).AND.(PZ.EQ.KN)) THEN
C
C  Set reference point values
C
               IK = IN + 1
               IL = IK
               JK = JN + 1
               JL = JK
               KK = KN + 1
               KL = KK
C
C  Initialize value for array B
C
               IC = 1
C
```

```
C  Examine the 27 grid blocks surrounding the grid block of choice (PX,PY,PZ)
C  and place the electrical or thermal information for these grid blocks in
C  the one dimensional array, B.
C
                DO 15 KA = KM,KK
                   DO 15 JA = JM,JK
                      DO 15 IA = IM,IK
                        IF ((IA.GT.NX).OR.(JA.GT.NY).OR.(KA.GT.NZ)) THEN
                           B(IC) = 0
                           GOTO 14
                        ELSE
                           B(IC) = A2(IA,JA,KA)
                        END IF
  14                     IC    = IC + 1
  15            CONTINUE
C
C  The array H provides a starting point within B at which groups of eight
C  grid blocks may be examined to determine if they can be used for
C  interpolation.
C
                H(1) =  2
                H(2) =  4
                H(3) = 10
                H(4) =  5
                H(5) = 13
                H(6) = 11
                H(7) = 14
C
C  The following subroutines determine which group of eight grid blocks would
C  be best for the interpolation.  Once the blocks have been chosen, the
C  program advances to the actual interpolation equation.
C
                CALL CHECK(B,H,FLAG)
                CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
                GOTO 11
             END IF
C
C  The following is similar code for other possible values of PX,PY and PZ
C
```

```
           IF ((PX.EQ.IM).AND.(PY.EQ.JN).AND.(PZ.EQ.KN)) THEN
              IK = IM - 1
              IL = IN - 1
              JK = JN + 1
              JL = JK
              KK = KN + 1
              KL = KK
             IC = 1
              DO 25 KA = KM,KK
                 DO 25 JA = JM,JK
                    DO 25 IA = IK,IN
                       IF ((IA.LT.1).OR.(JA.GT.NY).OR.(KA.GT.NZ)) THEN
                          B(IC) = 0
                          GOTO 24
                       ELSE
                          B(IC) = A2(IA,JA,KA)
                       END IF
 24                    IC    = IC + 1
 25           CONTINUE
              H(1) =  1
              H(2) =  5
              H(3) = 11
              H(4) =  4
              H(5) = 14
              H(6) = 10
              H(7) = 13
              CALL CHECK(B,H,FLAG)
              CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
              GOTO 11
           END IF
C
C
           IF ((PX.EQ.IM).AND.(PY.EQ.JM).AND.(PZ.EQ.KN)) THEN
              IK = IM - 1
              IL = IN - 1
              JK = JM - 1
              JL = JN - 1
              KK = KN + 1
              KL = KK
              IC = 1
```

```
               DO 35 KA = KM,KK
                 DO 35 JA = JK,JN
                    DO 35 IA = IK,IN
                       IF ((IA.LT.1).OR.(JA.LT.1).OR.(KA.GT.NZ)) THEN
                          B(IC) = 0
                          GOTO 34
                       ELSE
                          B(IC) = A2(IA,JA,KA)
                       END IF
 34                    IC    = IC + 1
 35           CONTINUE

              H(1) =  4
              H(2) =  2
              H(3) = 14
              H(4) =  1
              H(5) = 11
              H(6) = 13
              H(7) = 10
              CALL CHECK(B,H,FLAG)
              CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
              GOTO 11
            END IF
C
C
           IF ((PX.EQ.IN).AND.(PY.EQ.JM).AND.(PZ.EQ.KN)) THEN
              IK = IN + 1
              IL = IK
              JK = JM - 1
              JL = JN - 1
              KK = KN + 1
              KL = KK
              IC = 1
               DO 45 KA = KM,KK
                 DO 45 JA = JK,JN
                    DO 45 IA = IM,IK
                       IF ((IA.GT.NX).OR.(JA.LT.1).OR.(KA.GT.NZ)) THEN
                          B(IC) = 0
                          GOTO 44
```

```
                  ELSE
                     B(IC) = A2(IA,JA,KA)
                  END IF
  44              IC    = IC + 1
  45         CONTINUE

         H(1) =  5
         H(2) =  1
         H(3) = 13
         H(4) =  2
         H(5) = 10
         H(6) = 14
         H(7) = 11
         CALL CHECK(B,H,FLAG)
         CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
         GOTO 11
      END IF
C
C
      IF ((PX.EQ.IN).AND.(PY.EQ.JN).AND.(PZ.EQ.KM)) THEN
         IK = IN + 1
         IL = IK
         JK = JN + 1
         JL = JK
         KK = KM - 1
         KL = KN - 1
         IC = 1
         DO 55 KA = KK,KN
            DO 55 JA = JM,JK
               DO 55 IA = IM,IK
                 IF ((IA.GT.NX).OR.(JA.GT.NY).OR.(KA.LT.1)) THEN
                    B(IC) = 0
                    GOTO 54
                 ELSE
                    B(IC) = A2(IA,JA,KA)
                 END IF
  54              IC    = IC + 1
  55     CONTINUE

         H(1) = 11
```

```
          H(2) = 13
          H(3) =  1
          H(4) = 14
          H(5) =  4
          H(6) =  2
          H(7) =  5
          CALL CHECK(B,H,FLAG)
          CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
          GOTO 11
        END IF
C
C
        IF ((PX.EQ.IM).AND.(PY.EQ.JN).AND.(PZ.EQ.KM)) THEN
          IK = IM - 1
          IL = IN - 1
          JK = JN + 1
          JL = JK
          KK = KM - 1
          KL = KN - 1
         IC = 1
          DO 65 KA = KK,KN
            DO 65 JA = JM,JK
              DO 65 IA = IK,IN
                IF ((IA.LT.1).OR.(JA.GT.NY).OR.(KA.LT.1)) THEN
                  B(IC) = 0
                  GOTO 64
                ELSE
                  B(IC) = A2(IA,JA,KA)
                END IF
 64             IC    = IC + 1
 65       CONTINUE

          H(1) = 10
          H(2) = 14
          H(3) =  2
          H(4) = 13
          H(5) =  5
          H(6) =  1
          H(7) =  4
         CALL CHECK(B,H,FLAG)
         CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
         GOTO 11
      END IF
C
C
      IF ((PX.EQ.IM).AND.(PY.EQ.JM).AND.(PZ.EQ.KM)) THEN
         IK = IM - 1
         IL = IN - 1
         JK = JM - 1
         JL = JN - 1
         KK = KM - 1
         KL = KN - 1
         IC = 1
          DO 75 KA = KK,KN
            DO 75 JA = JK,JN
               DO 75 IA = IK,IN
                  IF ((IA.LT.1).OR.(JA.LT.1).OR.(KA.LT.1)) THEN
                     B(IC) = 0
                     GOTO 74
                  ELSE
                     B(IC) = A2(IA,JA,KA)
                  END IF
 74               IC    = IC + 1
 75      CONTINUE

         H(1) = 13
         H(2) = 11
         H(3) =  5
         H(4) = 10
         H(5) =  2
         H(6) =  4
         H(7) =  1
         CALL CHECK(B,H,FLAG)
         CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
         GOTO 11
      END IF
C
C
```

```
            IF ((PX.EQ.IN).AND.(PY.EQ.JM).AND.(PZ.EQ.KM)) THEN
                IK = IN + 1
                IL = IK
                JK = JM - 1
                JL = JN - 1
                KK = KM - 1
                KL = KN - 1
                IC = 1
                 DO 85 KA = KK,KN
                   DO 85 JA = JK,JN
                      DO 85 IA = IM,IK
                         IF ((IA.GT.NX).OR.(JA.LT.1).OR.(KA.LT.1)) THEN
                            B(IC) = 0
                            GOTO 84
                         ELSE
                            B(IC) = A2(IA,JA,KA)
                         END IF
  84                     IC    = IC + 1
  85            CONTINUE

                H(1) = 14
                H(2) = 10
                H(3) =  4
                H(4) = 11
                H(5) =  1
                H(6) =  5
                H(7) =  2
                CALL CHECK(B,H,FLAG)
                CALL PICK(FLAG,IL,JL,KL,IN,JN,KN)
                GOTO 11
             END IF
C
C
  11         CONTINUE
C
C Calculation of values necessary for the interpolation.
C
             IM = IN - 1
             JM = JN - 1
             KM = KN - 1
            XL    = XT(I)       - XCOORD(IM)
            XR    = XCOORD(IN)  - XT(I)
            XBASE = XR          + XL
            YL    = YT(J)       - YCOORD(JM)
            YR    = YCOORD(JN)  - YT(J)
            YBASE = YR          + YL
            ZL    = ZT(K)       - ZCOORD(KM)
            ZR    = ZCOORD(KN)  - ZT(K)
            ZBASE = ZR          + ZL
C
C  Trilinear interpolation
C
        A3(I,J,K) = (ZL*(YL * (XR*A(IM,JN,KN) + XL*A(IN,JN,KN)) +
     #                   YR * (XR*A(IM,JM,KN) + XL*A(IN,JM,KN))) +
     #               ZR*(YL * (XR*A(IM,JN,KM) + XL*A(IN,JN,KM)) +
     #                   YR * (XR*A(IM,JM,KM) + XL*A(IN,JM,KM))))
     #              /(XBASE*YBASE*ZBASE)
C
C
  1     CONTINUE
        RETURN
         END
C
C
C
       SUBROUTINE CHECK(B,H,FLAG)
C
C  This subroutine will determine if there are any electrically or thermally
C  insulating blocks in a given group of eight grid points.  This is done by
C  multiplying together the values used to describe the electrical or thermal
C  properties of each grid block.  Such a value will be equal to zero if a
C  block is insulating thus the product of the multiplication will also be
C  zero if one or more of the eight blocks is insulating.  The routine will
C  systematically test the seven different groups of eight grid blocks until
C  a nonzero product is obtained.  If the product always equals zero, a default
C  value is chosen whereby the original eight grid blocks are used in the
C  interpolation even though one or more of these blocks may be insulating.
C
C    B    - the array which stores the values representing the thermal or
```

```
C             electrical properties of the 27 grid blocks immediately surrounding
C             the block where the desired point is located.
C     H     - the array which stores the starting point value for each of the
C             seven possible groups of eight interpolation points.
C     FLAG - integer whose value determines which group of interpolation points
C             is used
C
         INTEGER AA(7),B(27),H(7),FLAG
C
         DO 51 G=1,7
            FLAG = G
            AA(G) = B(H(G)) * B(H(G)+1) * B(H(G)+3) * B(H(G)+4) * B(H(G)+9)
     #              * B(H(G)+10) * B(H(G)+12) * B(H(G)+13)
            IF(AA(G).NE.0) GOTO 52
   51    CONTINUE
         FLAG = 8
   52    CONTINUE
         RETURN
         END
C
C
C
         SUBROUTINE PICK(F,I,J,K,IN,JN,KN)
C
C  Based on the value of FLAG that is returned from subroutine CHECK, this
C  subroutine will adjust the array coordinates so that the proper values
C  will be used in the interpolation.
C
C     F         - equal to the value of FLAG in subroutine CHECK
C     I,J,K     - new reference points for array coordinates
C     IN,JN,KN - reference points for array coordinates used in the
C                 interpolation
C
         INTEGER F,I,J,K,IN,JN,KN
C
         IF (F.EQ.1) THEN
            IN = I
         ELSE IF (F.EQ.2) THEN
            JN = J
         ELSE IF (F.EQ.3) THEN
   KN = K
ELSE IF (F.EQ.4) THEN
   IN = I
   JN = J
ELSE IF (F.EQ.5) THEN
   JN = J
   KN = K
ELSE IF (F.EQ.6) THEN
   IN = I
   KN = K
ELSE IF (F.EQ.7) THEN
   IN = I
   JN = J
   KN = K
END IF
RETURN
END
```

Version 1.3

When the EPEIOS program was originally conceived on the University of Alberta mainframe computer, graphical output was produced by delivering data to graphics programs that resided on MTS. Since the exact same graphics programs were not available on the Macintosh system, the method used to generate graphical output had to be revised.

The desired graphing capabilities were to be able to generate contour plots, greyscale plots and 3-D plots. Since a graphics package that could meet these criteria was already available on the Macintosh system in the form of SYSTAT 5.0, the EPEIOS program was modified such that it was compatible with SYSTAT.

Compatibility was achieved by generating an output data file in tab delimited format. Data files in this format can be imported directly into SYSTAT. For convenience and versatility, variables that dictate the form of the output data were added to the program. These variables can be defined by the user in the input namelist &INWELL.

An outline of the changes that were made are as follows:

- in subroutine INPUT, output file EPEIOS.GRAPH:/DATA was assigned to I/O unit IOGRAP (IOGRAP=3)
- in subroutine OUTPUT, code was added that would write data to EPEIOS.GRAPH:/DATA in tab delimited columns for the cartesian plane specified by the user. Data columns are displayed in the following order: Time, X coord., Y coord., Z coord., Temperature, Real Electric Potential, Imaginary Electric Potential and Heating Rate.
- the following variables were added to input namelist &INWELL and to common block COMCON

  LGRAPH - a logical variable which governs printing of graphing data. If LGRAPH is set equal to .TRUE., data is printed in /DATA. If LGRAPH is set equal to .FALSE., printing of data is suppressed. The default value for this variable is .FALSE..

PLANE - specifies the cartesian plane in which the values of the variables to be graphed are calculated. This variable may be specified as being equal to 'XY', 'YZ' or 'XZ'.

CUTS - specifies the position of the chosen plane along the axis running perpendicular to it with respect to the origin. Several positions may be specified (maximum of five).

NCUT - specifies how many different values were given in CUTS. NCUT is an integer value that is less than or equal to five.

- the variables POTRG, POTIG, QG and TEMPG were created to store the graphing data that is generated.

- the file EPEIOS.Scripts:Clearout was modified so that the file EPEIOS.GRAPH:/DATA would also be emptied when the script 'clearout' was executed.

- in subroutine INPUT, the variable VERSN was set equal to 1.3.

Version 1.4

Given that EPEIOS generates data for three dimensional configurations, the ability to portray the data graphically in three dimensions becomes an important feature. Although SYSTAT was easy to implement and able to provide many forms of graphical output, it was found to have limited 3-D capabilities (no isosurfaces). In order to produce the desired 3-D graphs, a new graphics package had to be used. At the time, the best available package was a public domain graphics program produced by NCSA called Image 3.0.

This program was able to generate the required graphics but it could only read in data files that were in a special format called Hierarchical Data Format (HDF). In order to generate a data file in HDF, further changes had to be made to EPEIOS. Most of the modifications to the code were performed by Scott Juskiw as a representative for the Apple Research Partnership Program. In connection with the changes made to the code, revisions to the file structure (directories) and other subroutines were also carried out.

To generate data in HDF, two new subroutines and a common block were added to the EPEIOS code. With the help of the trilinear interpolation program LININT, these subroutines employed the user defined value for data point spacing (MAXCUBE) to produce a separate dataset in HDF for real electric potential, imaginary electric potential, heating rate and temperature. New datasets were generated whenever a printed output was requested. For convenience, all HDF datasets were appended into the same file.

Shortly after most modifications had been completed, a new 3-D graphics package called Spyglass was acquired. Spyglass is slighty more advanced than Image in its graphical displays and like Image it is set up to read data files that are in HDF. However, unlike Image 3.0, Spyglass is only capable of reading in one dataset per file. To accommodate both packages, a new approach had to be taken whereby a different file was created for each HDF dataset. Each HDF file created is located in the folder HDFFORM found in EPEIOS:RUNTIME.I/O.

An outline of the changes that were made are as follows:

- A new folder was created called RUNTIME.I/O which contains the files that may be read from or written to during a specific run (OUTPUT, TABFORM (formerly called DATA), RESTART and STORE).

- created subroutine INITHDF which contains initialization and formatting specifications for dataset labels used in the HDF files. This subroutine also takes the user defined value of MAXCUBE and generates an array of data points within the defined grid structure that are as equally spaced in each dimension as possible. Equal spacing between data points is necessary to produce an accurate three dimensional graphical image of the data.

- created subroutine MAKEHDF which calls upon subroutine LININT to determine the data values at the points defined in INITHDF. The data array resulting from the interpolation is called the scientific dataset (SDS). A unique file name is created for each dataset based on the type of data and the timestep number and the SDS is then stored in that file.

- created common block file COMHDF which contains the new variables used in subroutines INITHDF and MAKEHDF.

- introduce two new variables into the input namelist:
    - LHDF - a logical variable with a default value of FALSE. When it is set to TRUE, an HDF file will be generated.
    - MAXCUBE - an integer variable whose value defines the spacing of the data points (used for generating a 3-D graph) in the direction corresponding to the longest dimension of the configuration.

- modify EPEIOS.make by adding the libraries that are necessary to generate a data file in HDF.

- in subroutines INHEAT, MAIN, OUTPUT and LININT, include the common block COMHDF.

- in subroutine INHEAT, initialize LHDF to FALSE.

- in subroutine MAIN, insert call to INITHDF if LHDF is set to TRUE.

- in subroutine OUTPUT, insert a call to MAKEHDF if LHDF is set to TRUE.

- in subroutine LININT, array declaration for variable A3 was changed so that it is dependent on the array dimensions that are passed to it in the call statement.

- in subroutine OUTARR and OUTARI the array declarations of all arrays were changed so that they are dependent on the array dimensions that are passed in the call statement.

- in subroutine LININT, the variables NPX, NPY, and NPZ were changed to NOX, NOY and NOZ respectively. Change was made to avoid multiple uses of the same variable name.

- in subroutine LININT, code was added that would determine the maximum and minimum values in the array of interpolated data points (AMAXVALUE and AMINVALUE).

- in subroutine OUTPUT, code that generated the tab delimited data file was removed and made into two separate subroutines called MAKETAB and DATADUMP. MAKETAB makes the necessary calls to the linear interpolation subroutine and then calls DATADUMP which prints out the results of the interpolation in tab delimited columns. Notice that the order of the columns has changed slightly to match the order of the data presented in the printed output (ie. time, x, y, z, real potential, imaginary potential, heating rate and temperature). Both of these subroutines are located in the file MAKETAB.F.

- the name of the file which the data in tab delimited columns was written to was changed from RUNTIME.I/O:DATA to RUNTIME.I/O:TABFORM.

- the logical variable which determines if data in tab delimited columns is generated was changed from LGRAPH to LTAB.

- in subroutine RUNTIM, modify code to compute the CPU time used in hours, minutes and seconds. Three new variables (CTH, CTM, and CTS) were added to the common block COMSYS.

- in subroutine OUTPUT, modify code so that the amount of CPU time used would be printed to the screen in the desired format (hours, minutes, seconds).

- rewrite the script CLEAROUT such that it performs the same function as before except more efficiently.

- in the User Startup file in MPW, define {EPEIOSFOLDER} as being equal to the directory path that defines the location of the EPEIOS folder (unique to each machine). All scripts which had explicit pathnames that specified the location of the EPEIOS folder were modified such that pathnames were specified by {EPEIOSFOLDER}.

- the file EPEIOS.Scripts was moved from the MPW folder to the EPEIOS folder and renamed SCRIPTS. This required that {EPEIOSFOLDER}SCRIPTS be added to the Set Commands in the User Startup file in MPW.

- in subroutine INPUT, the I/O unit specifications were changed such that the pathnames are not specific to any one machine (ie. hard drive is not explicitly named).

- using the program RESEDIT, a unique icon was created for the EPEIOS application.

- in subroutine INPUT, the variable VERSN was set equal to 1.4.

```
      SUBROUTINE INITHDF
C
C This subroutine initializes many of the labels and formats that are required
C in an HDF file.  It also establishes an array of data points based on the
C user defined value of MAXCUBE.
C
      INCLUDE ':COMMONS:COMGEO.F'
      INCLUDE ':COMMONS:COMHDF.F'

      INTEGER DFSDCLEAR,DFSDSETDIMS,DFSDSETDIMSCALE
      INTEGER HDFERR

      REAL*4 XWIDTH,YWIDTH,ZWIDTH,AMAXWIDTH,XINC,YINC,ZINC
C
C Designate the folder (directory path) where all the different HDF files
C that are written will be located.
C
      SDSFOLDER = '::RUNTIME.I/O:HDFFORM'
C
C Define the coordinate system that will be used and the number of dimensions
C (rank) that are described in the data.
C
      COORDSYS = 'CARTESIAN'
      RANK     =  3
C
C Initialize beginning reference number for data in the HDF file.
C
      REFNO = 1
C
C Initialize labels, units and formats for the spatial variables.  For
C generality, each dimension are given four different assignments for label,
C unit and format since there will four different values (real potential,
C imaginary potential, heating rate and temperature) at each data point.
C
      DIMLABEL(1,1) = 'X'
      DIMLABEL(1,2) = 'Y'
      DIMLABEL(1,3) = 'Z'
      DIMLABEL(2,1) = 'X'
      DIMLABEL(2,2) = 'Y'
      DIMLABEL(2,3) = 'Z'
      DIMLABEL(3,1) = 'X'
      DIMLABEL(3,2) = 'Y'
      DIMLABEL(3,3) = 'Z'
      DIMLABEL(4,1) = 'X'
      DIMLABEL(4,2) = 'Y'
      DIMLABEL(4,3) = 'Z'

      DIMUNIT(1,1) = 'METRES'
      DIMUNIT(1,2) = 'METRES'
      DIMUNIT(1,3) = 'METRES'
      DIMUNIT(2,1) = 'METRES'
      DIMUNIT(2,2) = 'METRES'
      DIMUNIT(2,3) = 'METRES'
      DIMUNIT(3,1) = 'METRES'
      DIMUNIT(3,2) = 'METRES'
      DIMUNIT(3,3) = 'METRES'
      DIMUNIT(4,1) = 'METRES'
      DIMUNIT(4,2) = 'METRES'
      DIMUNIT(4,3) = 'METRES'

      DIMFORMAT(1,1) = 'F9.4'
      DIMFORMAT(1,2) = 'F9.4'
      DIMFORMAT(1,3) = 'F9.4'
      DIMFORMAT(2,1) = 'F9.4'
      DIMFORMAT(2,2) = 'F9.4'
      DIMFORMAT(2,3) = 'F9.4'
      DIMFORMAT(3,1) = 'F9.4'
      DIMFORMAT(3,2) = 'F9.4'
      DIMFORMAT(3,3) = 'F9.4'
      DIMFORMAT(4,1) = 'F9.4'
      DIMFORMAT(4,2) = 'F9.4'
      DIMFORMAT(4,3) = 'F9.4'
C
C  Initialize labels, units and formats for real potential, imaginary potential
C  heating rate and temperature.
C
      DATALABEL(1) = 'REAL POT.'
      DATALABEL(2) = 'IMAG. POT.'
      DATALABEL(3) = 'HEAT RATE'
      DATALABEL(4) = 'TEMPERATURE'

      DATAUNIT(1) = 'VOLTS'
      DATAUNIT(2) = 'VOLTS'
      DATAUNIT(3) = 'WATTS/CUBIC METRE'
      DATAUNIT(4) = 'DEGREES CELSIUS'

      DATAFORMAT(1) = 'F9.4'
      DATAFORMAT(2) = 'F9.4'
      DATAFORMAT(3) = 'F9.4'
      DATAFORMAT(4) = 'F9.4'
C
C  Calculate the length of each side of the grid structure.
C
      XWIDTH = XCOORD(NX) + DELTAX(NX)/2 - XCOORD(1) + DELTAX(1)/2
      YWIDTH = YCOORD(NY) + DELTAY(NY)/2 - YCOORD(1) + DELTAY(1)/2
      ZWIDTH = ZCOORD(NZ) + DELTAZ(NZ)/2 - ZCOORD(1) + DELTAZ(1)/2
C
C  Determine which side is longest.
C
      AMAXWIDTH = AMAX1(XWIDTH,YWIDTH,ZWIDTH)
C
C  Use the value of MAXCUBE to construct a new set of data points which are as
C  equally spaced as possible.  MAXCUBE is defined by the user in the input
C  namelist and is equal to the number of data points that are taken along the
C  longest side of the original grid structure.  Equispaced data points are
C  necessary for an accurate three dimensionsal graphical representation of the
C  data.
C
      IF(AMAXWIDTH.EQ.XWIDTH) THEN
        NUMX = MAXCUBE
          XINC = XWIDTH/NUMX
        NUMY = INT(YWIDTH/XINC + 0.5)
          YINC = YWIDTH/NUMY
        NUMZ = INT(ZWIDTH/XINC + 0.5)
          ZINC = ZWIDTH/NUMZ

      ELSE IF(AMAXWIDTH.EQ.YWIDTH) THEN
        NUMY = MAXCUBE
```

```
            YINC = YWIDTH/NUMY
          NUMX = INT(XWIDTH/YINC + 0.5)
            XINC = XWIDTH/NUMX
          NUMZ = INT(ZWIDTH/YINC + 0.5)
            ZINC = ZWIDTH/NUMZ

        ELSE
          NUMZ = MAXCUBE
            ZINC = ZWIDTH/NUMZ
          NUMX = INT(XWIDTH/ZINC + 0.5)
            XINC = XWIDTH/NUMX
          NUMY = INT(YWIDTH/ZINC + 0.5)
            YINC = YWIDTH/NUMY
        END IF

        IDIMSIZE(1) = NUMX
        IDIMSIZE(2) = NUMY
        IDIMSIZE(3) = NUMZ
C
C  Find the center points of all the grid blocks formed by the redefined grid
C  structure.
C
        XPOS(1) = XINC/2
        YPOS(1) = YINC/2
        ZPOS(1) = ZINC/2

        DO 1 I=2,NUMX
 1            XPOS(I) = XINC+XPOS(I-1)
        DO 2 I=2,NUMY
 2            YPOS(I) = YINC+YPOS(I-1)
        DO 3 I=2,NUMZ
 3            ZPOS(I) = ZINC+ZPOS(I-1)
C
C  Clear all possible previously used 'set' commands (SDS routines).
C
        HDFERR = DFSDCLEAR()
         IF(HDFERR.NE.0) WRITE(IOUSER,100)
C
C  Establish rank and dimension sizes for subsequent data.
```

```
C
      HDFERR = DFSDSETDIMS(RANK,IDIMSIZE)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
C
C  Define a incrementation scale for each dimension using the equispaced
C  coordinates.
C
      HDFERR = DFSDSETDIMSCALE(1,IDIMSIZE(1),XPOS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSCALE(2,IDIMSIZE(2),YPOS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSCALE(3,IDIMSIZE(3),ZPOS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)

 100  FORMAT(' HDF error in INITHDF')

      RETURN
      END
```

SUBROUTINE MAKEHDF

```
        SUBROUTINE MAKEHDF
C
C  This subroutine uses the initializations performed in INITHDF to write
C  a data file in HDF.
C
      INCLUDE ':COMMONS:COMSYS.F'
      INCLUDE ':COMMONS:COMGLO.F'
      INCLUDE ':COMMONS:COMGEO.F'
      INCLUDE ':COMMONS:COMHDF.F'

      INTEGER DFSDSETMAXMIN,DFSDSETDIMSTRS,DFSDSETDATASTRS,DFSDPUTDATA
      INTEGER DFANPUTLABEL
      INTEGER HDFERR, TAG, DIVISOR, MANT, THISNUM

      REAL*4 HDFSDS(40,40,40)

      STRING*32 ANNOSTR(4),NUMSTR,SDSFOLDERSTR

      CHARACTER*1 NUMCHAR
      CHARACTER*32 ANNOCHAR(4)
C
C  Transfer the labels in DATALABEL into the string variable ANNOSTR
C
       ANNOSTR(1) = TRIM(DATALABEL(1))
       ANNOSTR(2) = TRIM(DATALABEL(2))
       ANNOSTR(3) = TRIM(DATALABEL(3))
       ANNOSTR(4) = TRIM(DATALABEL(4))
C
C  Transform the character variable SDSFOLDER (defined in subroutine INITHDF)
C  into a string variable.
C
       SDSFOLDERSTR = TRIM(SDSFOLDER)
C
C  Initialize
C
      NUMSTR = ''
      DIVISOR = 1
```

```
C
C  Set variable THISNUM equal to the timestep number (calculated in
C  subroutine MAIN).
C
      THISNUM = NSTEP
C
C  The following DO WHILE loops place the timestep number (THISNUM) into
C  the string NUMSTR.
C
      DO WHILE(DIVISOR.NE.0)
          IF(THISNUM.LT.(DIVISOR*10)) THEN
            LEAVE
        ELSE
            DIVISOR=DIVISOR*10
        END IF
      END DO

      DO WHILE(DIVISOR.NE.0)
          MANT = THISNUM/DIVISOR
        NUMCHAR = CHAR(MANT+48)
        NUMSTR = NUMSTR//NUMCHAR
        THISNUM=JMOD(THISNUM,DIVISOR)
        DIVISOR = DIVISOR/10
      END DO
C
C  Combine the time step number and the label from DATALABEL into a single
C  character label.
C
      ANNOCHAR(1) = ANNOSTR(1)//' '//NUMSTR
      ANNOCHAR(2) = ANNOSTR(2)//' '//NUMSTR
      ANNOCHAR(3) = ANNOSTR(3)//' '//NUMSTR
      ANNOCHAR(4) = ANNOSTR(4)//' '//NUMSTR
C
C  Specify the tag number for a scientific data group (NCSA defined).
C
      TAG = 700
C
C  Use trilinear interpolation to establish the value of real potential at
C  calculated data points.
```

```
C
      CALL LININT(POTR,ELEGEO,HDFSDS,XPOS,YPOS,ZPOS,NUMX,NUMY,NUMZ)
C
C  Create a unique file name in the folder specified by the variable
C  SDSFOLDERSTR by combining strings which describe data type (ANNOSTR) and
C  timestep number (NUMSTR).
C
      SDSFILENAME = SDSFOLDERSTR//':'//ANNOSTR(1)//' '//NUMSTR
C
C  Store highest and lowest values from the data array in the data file.
C
      HDFERR = DFSDSETMAXMIN(AMAXVALUE,AMINVALUE)
C
C  All scientific dataset routines are functions of integer type.  They will
C  return a value of zero if the function is successfully completed and a value
C  of -1 if it fails in some way.
C
      IF(HDFERR.NE.0) WRITE(IOUSER,100)
C
C  Specify label, unit and format for the x, y and z dimensions
C
      HDFERR = DFSDSETDIMSTRS(1,DIMLABEL(1,1),DIMUNIT(1,1),DIMFORMAT(1,1))
      IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(2,DIMLABEL(1,2),DIMUNIT(1,2),DIMFORMAT(1,2))
      IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(3,DIMLABEL(1,3),DIMUNIT(1,3),DIMFORMAT(1,3))
      IF(HDFERR.NE.0) WRITE(IOUSER,100)
C
C  Specify label, unit and format for raw data.
C
      HDFERR = DFSDSETDATASTRS(DATALABEL(1),DATAUNIT(1),DATAFORMAT(1),COORDSYS)
      IF(HDFERR.NE.0) WRITE(IOUSER,100)
C
C  Write data from HDFSDS to SDSFILENAME specifying number of dimensions
C  (rank) and the number of data points in each direction (IDIMSIZE).
C
      HDFERR = DFSDPUTDATA(SDSFILENAME,RANK,IDIMSIZE,HDFSDS)
      IF(HDFERR.NE.0) WRITE(IOUSER,100)
C
C  Assign a label to the data object specified by the tag and reference
```

```
C  number.
C
       HDFERR = DFANPUTLABEL(SDSFILENAME,TAG,REFNO,ANNOCHAR(1))
        IF(HDFERR.NE.0) WRITE(IOUSER,100)
C
C  Repeat process for imaginary potential, heating rate and temperature.
C
      CALL LININT(POTI,ELEGEO,HDFSDS,XPOS,YPOS,ZPOS,NUMX,NUMY,NUMZ)
      SDSFILENAME = SDSFOLDERSTR//':'//ANNOSTR(2)//' '//NUMSTR
      HDFERR = DFSDSETMAXMIN(AMAXVALUE,AMINVALUE)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(1,DIMLABEL(2,1),DIMUNIT(2,1),DIMFORMAT(2,1))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(2,DIMLABEL(2,2),DIMUNIT(2,2),DIMFORMAT(2,2))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(3,DIMLABEL(2,3),DIMUNIT(2,3),DIMFORMAT(2,3))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDATASTRS(DATALABEL(2),DATAUNIT(2),DATAFORMAT(2),COORDSYS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDPUTDATA(SDSFILENAME,RANK,IDIMSIZE,HDFSDS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
       HDFERR = DFANPUTLABEL(SDSFILENAME,TAG,REFNO,ANNOCHAR(2))
        IF(HDFERR.NE.0) WRITE(IOUSER,100)

      CALL LININT(Q,THMGEO,HDFSDS,XPOS,YPOS,ZPOS,NUMX,NUMY,NUMZ)
      SDSFILENAME = SDSFOLDERSTR//':'//ANNOSTR(3)//' '//NUMSTR
      HDFERR = DFSDSETMAXMIN(AMAXVALUE,AMINVALUE)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(1,DIMLABEL(3,1),DIMUNIT(3,1),DIMFORMAT(3,1))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(2,DIMLABEL(3,2),DIMUNIT(3,2),DIMFORMAT(3,2))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(3,DIMLABEL(3,3),DIMUNIT(3,3),DIMFORMAT(3,3))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDATASTRS(DATALABEL(3),DATAUNIT(3),DATAFORMAT(3),COORDSYS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDPUTDATA(SDSFILENAME,RANK,IDIMSIZE,HDFSDS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFANPUTLABEL(SDSFILENAME,TAG,REFNO,ANNOCHAR(3))
```

```
       IF(HDFERR.NE.0) WRITE(IOUSER,100)

      CALL LININT(TEMP,THMGEO,HDFSDS,XPOS,YPOS,ZPOS,NUMX,NUMY,NUMZ)
      SDSFILENAME = SDSFOLDERSTR//':'//ANNOSTR(4)//' '//NUMSTR
      HDFERR = DFSDSETMAXMIN(AMAXVALUE,AMINVALUE)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(1,DIMLABEL(4,1),DIMUNIT(4,1),DIMFORMAT(4,1))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(2,DIMLABEL(4,2),DIMUNIT(4,2),DIMFORMAT(4,2))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDIMSTRS(3,DIMLABEL(4,3),DIMUNIT(4,3),DIMFORMAT(4,3))
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDSETDATASTRS(DATALABEL(4),DATAUNIT(4),DATAFORMAT(4),COORDSYS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
      HDFERR = DFSDPUTDATA(SDSFILENAME,RANK,IDIMSIZE,HDFSDS)
       IF(HDFERR.NE.0) WRITE(IOUSER,100)
       HDFERR = DFANPUTLABEL(SDSFILENAME,TAG,REFNO,ANNOCHAR(4))
        IF(HDFERR.NE.0) WRITE(IOUSER,100)

100   FORMAT(' HDF error in MAKEHDF')

      RETURN
      END
```

COMMON BLOCK COMHDF

```
C/ MODULE COMHDF
        REAL*4 XPOS(40),YPOS(40),ZPOS(40),AMAXVALUE,AMINVALUE
        INTEGER*4 RANK,MAXCUBE,NUMX,NUMY,NUMZ,REFNO,IDIMSIZE(3)
        CHARACTER*64 SDSFILENAME
        CHARACTER*32 DIMLABEL(4,3),DIMUNIT(4,3),DIMFORMAT(4,3),SDSFOLDER,
     #               DATALABEL(4),DATAUNIT(4),DATAFORMAT(4),COORDSYS
        LOGICAL LHDF
        COMMON /COMHDF/
     #        XPOS,YPOS,ZPOS,AMAXVALUE,AMINVALUE,
     #        RANK,MAXCUBE,NUMX,NUMY,NUMZ,REFNO,IDIMSIZE,SDSFOLDER,
     #        DIMLABEL,DIMUNIT,DIMFORMAT,DATALABEL,DATAUNIT,DATAFORMAT,
     #        COORDSYS,LHDF
```

SUBROUTINE MAKETAB

```
  SUBROUTINE MAKETAB

   INCLUDE ':COMMONS:COMCON.F'
   INCLUDE ':COMMONS:COMGLO.F'
   INCLUDE ':COMMONS:COMGEO.F'
C
C  The data is calculated for a cross section of the retangular grid structure.
C  The user may choose in which plane (xy, yz or xz) he would like the cross
C  section.  The variable CUTS (specified by the user), determines the point(s),
C  relative to the origin, at which the cross section(s) will be taken.  The
C  trilinear interpolation subroutine is used to calculate the values for
C  potential, heating rate and temperature in a given plane. The variable
C  NCUT (specified by the user) determines how many cross sections (max of 5)
C  are desired in that plane.
C
       IF (PLANE.EQ.'XY') THEN
         CALL LININT(POTR,ELEGEO,POTRG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
         CALL LININT(POTI,ELEGEO,POTIG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
         CALL LININT(Q,THMGEO,QG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
         CALL LININT(TEMP,THMGEO,TEMPG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
         CALL DATADUMP(POTRG,POTIG,QG,TEMPG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
       END IF
       IF (PLANE.EQ.'YZ') THEN
         CALL LININT(POTR,ELEGEO,POTRG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
         CALL LININT(POTI,ELEGEO,POTIG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
         CALL LININT(Q,THMGEO,QG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
         CALL LININT(TEMP,THMGEO,TEMPG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
         CALL DATADUMP(POTRG,POTIG,QG,TEMPG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
       END IF
       IF (PLANE.EQ.'XZ') THEN
         CALL LININT(POTR,ELEGEO,POTRG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
         CALL LININT(POTI,ELEGEO,POTIG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
         CALL LININT(Q,THMGEO,QG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
         CALL LININT(TEMP,THMGEO,TEMPG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
         CALL DATADUMP(POTRG,POTIG,QG,TEMPG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
      END IF
     RETURN
```

```
      END
C
C Subroutine for printing out data in tab delimited columns
C
      SUBROUTINE DATADUMP(AR1,AR2,AR3,AR4,SL1,SL2,SL3,NC1,NC2,NC3)
C
      INCLUDE ':COMMONS:COMCON.F'
      INCLUDE ':COMMONS:COMSYS.F'
C
C Initialize all arrays
C
      REAL*4 AR1(NC1,NC2,NC3),AR2(NC1,NC2,NC3),AR3(NC1,NC2,NC3),AR4(NC1,NC2,NC3)
      REAL*4 SL1(NC1),SL2(NC2),SL3(NC3)
C
C Define the tab as a variable
C
      CHARACTER*1 TAB
      TAB = CHAR(9)
C
C Print out the data to unit IOGRAP, grouping the data according to the
C different slices (CUTS) for the given plane.
C
      IF (PLANE.EQ.'YZ') THEN
       DO 1 I=1,NC1
         DO 1 J=1,NC2
           DO 1 K=1,NC3
           WRITE(IOGRAP,4) TIME,TAB,SL1(I),TAB,SL2(J),TAB,SL3(K),TAB,AR1(I,J,K),
     #                     TAB,AR2(I,J,K),TAB,AR3(I,J,K),TAB,AR4(I,J,K)
   4       FORMAT(F9.2,A,3(F7.3,A),3(F9.2,A),F9.2)
   1   CONTINUE
      ELSE IF (PLANE.EQ.'XZ') THEN
       DO 2 J=1,NC2
         DO 2 K=1,NC3
           DO 2 I=1,NC1
           WRITE(IOGRAP,5) TIME,TAB,SL1(I),TAB,SL2(J),TAB,SL3(K),TAB,AR1(I,J,K),
     #                     TAB,AR2(I,J,K),TAB,AR3(I,J,K),TAB,AR4(I,J,K)
   5       FORMAT(F9.2,A,3(F7.3,A),3(F9.2,A),F9.2)
   2   CONTINUE
      ELSE IF (PLANE.EQ.'XY') THEN
```

```
          DO 3 K=1,NC3
            DO 3 J=1,NC2
              DO 3 I=1,NC1
              WRITE(IOGRAP,6) TIME,TAB,SL1(I),TAB,SL2(J),TAB,SL3(K),TAB,AR1(I,J,K),
     #                        TAB,AR2(I,J,K),TAB,AR3(I,J,K),TAB,AR4(I,J,K)
    6         FORMAT(F9.2,A,3(F7.3,A),3(F9.2,A),F9.2)
    3     CONTINUE
        END IF
C
C
         RETURN
         END
```

Version 15

Extended use of EPEIOS on a wide variety of configurations revealed several aspects of the program which could be improved upon. Most of the resulting changes fell into the category of program I/O as an effort was made to improve the flow of data to and from the application. Program control was altered with the introduction of a new namelist variable (logical) which when set will terminate the simulation when a maximum temperature of 100°C is reached within the heated formation. Program capability was expanded when the limitation on the size of grid structure (number of grid blocks in each direction) was increased from 40*40*40 to 100*100*100. In addition, the restriction on the number of regions that could be specified in a simulation was also increased from 40 to 100. Although these expansions allow for grid structures that would be too large for a Macintosh computer to handle, they are needed for situations where many grid blocks are required in one or two directions and very few are needed in the remaining direction(s) (i.e. a two dimensional problem). The maximum grid structure size that can be handled will be dictated by the amount of RAM that is available and the amount of time that one has to wait for results.

The EPEIOS code was also modified such that data files for graphing results along a two dimensional slice through the modelled configuration were generated in the ASCII X-Y format. Data in this format consists of any number of tab delimited columns with the two dimensional coordinates appearing in the first two columns. This format was compatible with a newer version of Transform which is the two dimensional graphing application in the Spyglass package. The modifications were completed in such a manner that the three dimensional nature of the data was preserved and compatibility with Systat was maintained.

An outline of the changes that were made is as follows:

- in subroutine TEST, code was added that would cause simulation progress data to be printed to the screen during run time for each timestep in which a printed output record is not generated.

- in subroutine OUTPUT, modifications were done to the message that is printed to the screen during run time when a printed output record is generated.

- in subroutine OSTORE, previously used format specifications in all formatted write statements to the data storage file, were replaced with the ' * ' format.

- in subroutine INREC, previously used format specifications in all formatted read statements from the restart record were replaced with the ' * ' formats.

- in subroutine INPUT, the default value for CPUMAX was changed from 1e5 seconds to 2.0e6 seconds (approximately 23 days).

- in subroutine INPUT, OPEN statements were relocated within the subroutine such that files would only be opened if data was to be written to them.

- in subroutines THOMAS and TSOLVE, all one dimensional array initializations were changed from 40 to 100.

- graphing variables (LTAB, PLANE, NCUT, CUTS, LHDF and MAXCUBE) were all moved from namelist &INWELL to namelist &INPUT3. Consequently, the variables were removed from the namelist declaration for &INWELL in subroutine INHEAT and added to the namelist declaration for INPUT3 in subroutine INPUT. In addition, the initialization of default values were moved from subroutine INHEAT and placed in subroutine INPUT. As a result of this modification, the inclusion of the common block COMHDF.F was no longer required in subroutine INHEAT so it was deleted.

- a new logical variable called MAXTEMP was added to namelist &INWELL and included in common block COMCON.F. If set to TRUE, the simulation will terminate when maximum temperature (TMAX) within a formation reaches 100°C.

- in subroutine INHEAT, default value of MAXTEMP was set to FALSE.

- in subroutine TEST, code was added that will test the value of TMAX and cause the simulation to end if it exceeds 100 and MAXTEMP is equal to TRUE.

- subroutine DATADUMP in the file MAKETAB.F was rewritten such that data files would be generated in ASCII X-Y format. This involved creating an additional subroutine (CONVERT) within the file MAKETAB.F. Refer to the program listing for further details.

- in subroutine INPUT, the OPEN statement which established a connection between an output unit and a specific file (:RUNTIME.I/O:TABFORM) was removed.

- in subroutine INPUT, the variable VERSN was set equal to 1.5.

```
      SUBROUTINE MAKETAB

      INCLUDE ':COMMONS:COMCON.F'
      INCLUDE ':COMMONS:COMGLO.F'
      INCLUDE ':COMMONS:COMGEO.F'
C
C   The data is calculated for a cross section of the retangular grid structure.
C   The user may choose in which plane (xy, yz or xz) he would like the cross
C   section.  The variable CUTS (specified by the user), determines the point(s),
C   relative to the origin, at which the cross section(s) will be taken.  The
C   trilinear interpolation subroutine is used to calculate the values for
C   potential, heating rate and temperature in a given plane. The variable
C   NCUT (specified by the user) determines how many cross sections (max of 5)
C   are desired in that plane.
C
          IF (PLANE.EQ.'XY') THEN
            CALL LININT(POTR,ELEGEO,POTRG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
            CALL LININT(POTI,ELEGEO,POTIG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
            CALL LININT(Q,THMGEO,QG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
            CALL LININT(TEMP,THMGEO,TEMPG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
            CALL DATADUMP(POTRG,POTIG,QG,TEMPG,XCOORD,YCOORD,CUTS,NX,NY,NCUT)
          END IF
          IF (PLANE.EQ.'YZ') THEN
            CALL LININT(POTR,ELEGEO,POTRG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
            CALL LININT(POTI,ELEGEO,POTIG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
            CALL LININT(Q,THMGEO,QG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
            CALL LININT(TEMP,THMGEO,TEMPG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
            CALL DATADUMP(POTRG,POTIG,QG,TEMPG,CUTS,YCOORD,ZCOORD,NCUT,NY,NZ)
          END IF
          IF (PLANE.EQ.'XZ') THEN
            CALL LININT(POTR,ELEGEO,POTRG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
            CALL LININT(POTI,ELEGEO,POTIG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
            CALL LININT(Q,THMGEO,QG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
            CALL LININT(TEMP,THMGEO,TEMPG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
            CALL DATADUMP(POTRG,POTIG,QG,TEMPG,XCOORD,CUTS,ZCOORD,NX,NCUT,NZ)
         END IF
       RETURN
       END
C
C  Subroutine for printing out data in tab delimited columns.  Data is subdivided
C  into several different files.  A new file is generated for each time data is
C  requested and for each data slice.  The time step number and the data slice
C  are recorded in the file name.  Each file will consist of eight columns of
C  data which appear in the following order.  The first three columns contain
```

```
C  the coordinates of the data points. The order in which these coordinates
C  appear depends on the plane that is chosen for the data slice. In all cases
C  the values in the third column will constant for each file.  The fourth
C  column contains the time at which the data was printed (constant for each
C  file).  Columns five through eight contain the following data: real potential,
C  imaginary potential, heating rate and temperature respectively.  The
C  dimensions of the data file (columns x rows) was also recorded in the file
C  name.
C
      SUBROUTINE DATADUMP(AR1,AR2,AR3,AR4,SL1,SL2,SL3,NC1,NC2,NC3)

      INCLUDE ':COMMONS:COMCON.F'
      INCLUDE ':COMMONS:COMSYS.F'
C
C  Initialize all arrays.
C
      REAL*4 AR1(NC1,NC2,NC3),AR2(NC1,NC2,NC3),AR3(NC1,NC2,NC3),AR4(NC1,NC2,NC3)
      REAL*4 SL1(NC1),SL2(NC2),SL3(NC3)
C
C  Define variable types.
C
      INTEGER ROWS,TNUM
      CHARACTER*1 TAB
      CHARACTER*16 FILE1,FILE2,FILE3,FILE4,FILE5,FILE6,FILE7
      CHARACTER*32 ROOT
      CHARACTER*64 PATH

      TAB = CHAR(9)
C
C  Define the folder in which the data files will be placed.
C
      ROOT = ':RUNTIME.I/O:TABFORM:'
C
C  Determine how many characters are present within the character variable 'ROOT'.
C
      LR = INDEX(ROOT,' ') - 1
C
C  Define the first part of the file name.
C
      FILE1 = 'ts='
      LF1 = INDEX(FILE1,'=')
C
C  Define the second part of the file name as the time step number.  This
C  requires using the subroutine CONVERT to change the integer variable NSTEP
C  into a character variable (FILE2) which can then be added to the file name.
```

```
C
      TNUM = NSTEP
      CALL CONVERT(TNUM,FILE2,LF2)
C
C  Define the fifth part of the file name. This part includes indication of the
C  number of columns that appear in each data file.
C
      FILE5 = ' (8x'
      LF5 = INDEX(FILE5,'x')
C
C  Define the seventh part of the file name.
C
      FILE7 = ')'
      LF7 = INDEX(FILE7,')')

C
C  Print out the data to unit IOGRAP, grouping the data according to the
C  different slices (CUTS) for the given plane.
C

      IF (PLANE.EQ.'YZ') THEN
C
C  Define the third part of the file name as the data slice.
C
        FILE3 = ',YZ'
        LF3 = INDEX(FILE3,'Z')
C
C  Calculate the number of rows that the tab delimited data file would contain.
C
        ROWS = NC2 * NC3
C
C  Change the integer value for number of rows into a character variable (FILE6)
C  which can be added to the file name.
C
        CALL CONVERT(ROWS,FILE6,LF6)
C
C  Loop through the data slices that are requested.
C
        DO 1 I=1,NC1
C
C  Change the data slice number into a character variable.  The number will
C  correspond to the order in which the data slices were requested in the
C  input namelist.
C
        FILE4 = CHAR(I+48)
      LF4 = 1
C
C  Combine all the parts of the file name into one character variable.
C
      PATH = ROOT(1:LR)//FILE1(1:LF1)//FILE2(1:LF2)//FILE3(1:LF3)
     #        //FILE4(1:LF4)//FILE5(1:LF5)//FILE6(1:LF6)//FILE7(1:LF7)
C
C  Establish a connection between the output unit and the generated file name.
C
      OPEN (IOGRAP,FILE=PATH)
C
C  Print out data in tab delimited columns.
C
        DO 1 K=1,NC3
          DO 1 J=1,NC2
          WRITE(IOGRAP,100) SL2(J),TAB,SL3(K),TAB,SL1(I),TAB,TIME,TAB,
     #                      AR1(I,J,K),TAB,AR2(I,J,K),TAB,AR3(I,J,K),TAB,
     #                      AR4(I,J,K)
 100      FORMAT(3(F7.3,A),E10.4,A,3(F9.2,A),F9.2)
   1  CONTINUE
C
C  Determine file name and print data if data slices in the XZ plane are
C  requested.
C
     ELSE IF (PLANE.EQ.'XZ') THEN

      FILE3 = ',XZ'
      LF3 = INDEX(FILE3,'Z')

      ROWS = NC1 * NC3

      CALL CONVERT(ROWS,FILE6,LF6)

      DO 2 J=1,NC2

      FILE4 = CHAR(J+48)
      LF4 = 1

      PATH = ROOT(1:LR)//FILE1(1:LF1)//FILE2(1:LF2)//FILE3(1:LF3)
     #        //FILE4(1:LF4)//FILE5(1:LF5)//FILE6(1:LF6)//FILE7(1:LF7)
      OPEN (IOGRAP,FILE=PATH)

        DO 2 K=1,NC3
         DO 2 I=1,NC1
         WRITE(IOGRAP,200) SL1(I),TAB,SL3(K),TAB,SL2(J),TAB,TIME,TAB,
     #                      AR1(I,J,K),TAB,AR2(I,J,K),TAB,AR3(I,J,K),TAB,
     #                      AR4(I,J,K)
 200        FORMAT(3(F7.3,A),E10.4,A,3(F9.2,A),F9.2)
   2    CONTINUE
C
C  Determine file name and print data if data slices in the XY plane are
C  requested.
C
      ELSE IF (PLANE.EQ.'XY') THEN

        FILE3 = ',XY'
        LF3 = INDEX(FILE3,'Y')

        ROWS = NC1 * NC2

        CALL CONVERT(ROWS,FILE6,LF6)

        DO 3 K=1,NC3

        FILE4 = CHAR(K+48)
        LF4 = 1

        PATH = ROOT(1:LR)//FILE1(1:LF1)//FILE2(1:LF2)//FILE3(1:LF3)
     #        //FILE4(1:LF4)//FILE5(1:LF5)//FILE6(1:LF6)//FILE7(1:LF7)
        OPEN (IOGRAP,FILE=PATH)

          DO 3 J=1,NC2
            DO 3 I=1,NC1
            WRITE(IOGRAP,300) SL1(I),TAB,SL2(J),TAB,SL3(K),TAB,TIME,TAB,
     #                      AR1(I,J,K),TAB,AR2(I,J,K),TAB,AR3(I,J,K),TAB,
     #                      AR4(I,J,K)
 300        FORMAT(3(F7.3,A),E10.4,A,3(F9.2,A),F9.2)
   3    CONTINUE
      END IF

      RETURN
      END
C
C  This subroutine converts an integer variable of unknown magnitude into a
C  character variable.  It receives the variable INUM and passes back the
C  variables CNUM and NT.
C
      SUBROUTINE CONVERT(INUM,CNUM,NT)

      INTEGER INUM,DIV,NT,MANT
      CHARACTER*1 NUMT
      CHARACTER*16 CNUM
C
C  Initialize the divisor.
C
      DIV = 1
C
C  Initialize a counter which keeps track of the number of digits in the
C  integer variable.
C
      NT = 1
C
C  Determine the relative magnitude of the integer variable and count the
C  number of digits.
C
    1 IF (DIV.NE.0) THEN
        IF (INUM.LT.(DIV*10)) THEN
          GOTO 2
        ELSE
          NT = NT+1
          DIV = DIV*10
          GOTO 1
        END IF
      END IF
    2 CONTINUE
C
C  Convert each digit in the integer variable (from left to right) into a
C  character and combine all characters into the character variable CNUM.
C
      DO 3 IT=1,NT
        MANT = INUM/DIV
        NUMT = CHAR(MANT+48)
        CNUM(IT:IT) = NUMT
        INUM = MOD(INUM,DIV)
        DIV = DIV/10
    3 CONTINUE
      RETURN
      END
```

Version 1.5 RP88

With the expansion of the limitation on the number of grid blocks that can be handled by EPEIOS, more complex and detailed simulations could be performed. After attempting some such simulations on the SE/30, it became apparent that more computing power was necessary to effectively utilize the capabilities of EPEIOS.

The most feasible solution was to transfer a copy of the EPEIOS program to a Macintosh II computer which had installed in it an RP88 coprocessor board. The RP88 is a high performance RISC processor that was developed by Tektronix. Previous use of the RP88 for running MAGAERA simulations showed that run times were reduced by a factor of ~20 compared to the SE/30 with a math coprocessor.

FORTRAN programs are compiled and executed on the RP88 board using an Absoft FORTRAN 77 compiler. This compiler is somewhat different from the Language Systems compiler used on the SE/30 in that it only supports strict FORTRAN 77. As a result, some of the EPEIOS code that utilized embellishments supported by the Language Systems compiler had to be replaced before the EPEIOS program could be made to compile and run on the RP88.

Modifications to the EPEIOS code were completed such that the entire program would compile with the Absoft compiler. Unfortunately, available libraries which support HDF functions were designed for compilation by the Language Systems compiler and could not easily be made to compile with the Absoft compiler. Without these libraries, data files in HDF cannot be generated. Consequently, code relating to the generation of HDF files was disabled in this version of the EPEIOS program even though it could be made to compile.

An outline of the all changes made to the EPEIOS code are as follows:

- in common block COMCON, variables NCUT and CUTS were placed ahead of the character variable PLANE.

- in common block file COMGLO, variable arrays QG, TEMPG, POTRG and POTIG were specified in the common block. Previously these variable arrays were declared as REAL*4 in the file but not included in the common block.

- in common block COMCON, a missing comma was placed after the variable MAXTEMP.

- in common block file COMHDF, character variable SDSFILENAME was included in the common block.

- in subroutine MAIN, a program statement was inserted in the first line to indicate the starting point of the program.

- in subroutine INPUT, the syntax of implied do loops used for writing coordinates and the thicknesses of grid blocks to the printed output file were made compatible with the new compiler.

- in subroutine MAKEHDF, all string variables were replaced by equivalent character variables.

- in the make file, compilation instructions involving HDF subroutines (INITHDF and MAKEHDF), the HDF common block (COMHDF) and linking instructions for the HDF libraries were commented out.

- in subroutines INPUT, LININT, OUTPUT and MAIN, all references to HDF subroutines, common blocks and variables were commented out.

- in subroutine RUNTIM, the call to the Language Systems subroutine SECNDS was replaced by a call to the RP88 subroutine TIMES. Previously used code was commented out.

- in subroutine INPUT, the name and location of the input namelist was specified as :RUNTIME.I/O:INPUT. This was a replacement of previous code which, under the Language Systems compiler, would open a dialog box and allow the user to select the input file from a specific folder.

APPENDIX C

EPEIOS User's Manual Version 1.5

Introduction

The purpose of this manual is to provide a reference guide to the program EPEIOS. It will be assumed that the reader is familiar with the finite difference method of finding approximate solutions to partial differential equations, and with the equations and approximations used in EPEIOS. Most of this information may be found in the Ph.D. thesis, "Computer Simulation of the Electric Preheat and Steam Drive Bitumen Recovery Process" written by Allan D. Hiebert.

The original program was developed by Hiebert on the University of Alberta mainframe computer system (MTS). It has since been modified and transferred to the Macintosh personal computer. Currently there are two variations of the EPEIOS program for the Macintosh. One is designed for running on any Macintosh with the Language Systems (LS) FORTRAN compiler (version 2.0). The other is designed for running on a high performance RP88 coprocessor designed by Tektronix which uses the Absoft FORTRAN 77 compiler. To distinguish between the two variations, the one designed for running on the RP88 will be referred to as EPEIOS88 and the other will be referred to as EPEIOS.LS. Unless otherwise specified, the information presented in this manual is common to both variations.

In using this manual it will be helpful if the reader is familiar with the Macintosh system, the Macintosh Programmer's Workshop (MPW) programming environment and the graphing packages Systat and Spyglass. Information on these subjects is available from the various user's manuals.

Namelist Data Input

The following namelists must be input, in order, into a single file. For further information on namelist input see "Language Systems FORTRAN 2.0 Reference Manual" pp 229 - 235.

Namelist &INPUT1

This namelist is used to input flow control and output labeling variables.

MNSTEP - ***Type: Integer, Default: None***
MNSTEP specifies the maximum number of timesteps for the simulation run. If the simulation has not been completed by the MNSTEP timestep, the run will terminate with a return code of 50. Before termination, a printed output (storage output and graphing data if requested) will be produced for the last timestep.

CPUMAX - ***Type:Real, Default:2.0E6, Units:seconds***
At the end of every timestep, the program checks how many CPU seconds have been used to that point in the run. If this number is greater than CPUMAX, then the run will be terminated with a return code of 50. No checking of CPU usage is done during the timestep calculation, so the actual time used by the run termination will be somewhat greater than CPUMAX. Before termination, a printed output (storage output and graphing data if requested) will be produced for the last timestep.

LRESTA - ***Type: Logical, Default: .FALSE.***
LRESTA must be set to .TRUE. for a restart run of a previously stored simulation run. By use of this control, the temperatures at some point in a previous run may be used as the start of the current run.

NREC - ***Type: Integer, Default: 1***
If LRESTA is set to .TRUE., the program will search the file containing the restart record for the storage record number NREC. The temperatures, etc., in this record will be used as the starting point for the current simulation run. When storage is specified for a simulation run (refer to the section on Namelist &INPUT3) the simulation time and record number of each stored record is recorded on the printed output of the simulation run.

LABEL1, LABEL2, LABEL3, LABEL4 - ***Type: Character, Default: Blank***

The user may specify zero to four labels to be printed at the start of the simulation (LABEL1, LABEL2, LABEL3, and LABEL4). These 48 character maximum labels may be used to describe the simulation, indicate who set up the simulation, record the sequence number of the simulation, etc.

Namelist &INPUT2

This namelist is used to input basic geometric parameters, such as the dimensions of the problem domain, the number of regions and electrodes, and the initial temperature.

NX - ***Type: Integer, Default: None***

NX specifies the number of grid blocks that the problem domain is divided into in the X direction. The problem domain must be composed of a regular array of rectangular prisms (grid blocks).

NY - ***Type: Integer, Default: None***

NY specifies the number of grid blocks that the problem domain is divided into in the Y direction.

NZ - ***Type: Integer, Default: None***

NZ specifies the number of grid blocks that the problem domain is divided into in the Z direction.

DELTAX - ***Type: Real Array, Default: None, Units: meters***

This array specifies the X dimensions of the grid blocks, starting with the left most plane of grid blocks. The X dimension of each Y-Z plane of grid blocks must be specified, so there must be at least NX elements in DELTAX.

DELTAY - ***Type: Real Array, Default: None, Units: meters***

This array specifies the Y dimensions of the X-Z planes of grid blocks. It must contain NY elements.

DELTAZ - ***Type: Real Array, Default: None, Units: meters***

This array specifies the Z dimension of the X-Y planes of grid blocks. It must contain NZ elements.

NREG	- ***Type: Integer, Default: None*** NREG specifies the number of different regions that are in the problem domain. It also indicates the number of &REGION namelists which follow immediately after namelist &INPUT2.
NELECT	- ***Type: Integer, Default: None*** NELECT specifies the number of different electrodes in the problem domain. This may be less than the number of electrode regions if two or more electrode regions (adjacent, overlapping, or separate) are used to form a single, non-rectangular, electrode.
TINIT	- ***Type: Real, Default: 24., Units: °C*** TINIT specifies the initial, uniform, temperature of the problem domain.
LPREG	- ***Type: Logical, Default: .TRUE.*** LPREG controls printing of region data to the output record. If LPREG is set to FALSE, characteristic data for each region will not be written in the printed output file.

Namelist &REGION

These namelists (of which there must be NREG in number) are used to specify the location and properties of different regions in the problem domain. All regions are rectangular prisms in shape, however, more complicated shapes may be formed by adding adjacent or overlapping regions with the same properties. The regions are lain down in consecutive order as they are read in, and thus, a later region may cover all or part of a previously read in region. The default region, Region 0, which initially contains the entire grid, is electrically and thermally insulating. Any portion of the grid which is not contained in one of the read in regions will remain electrically and thermally insulating.

The electrical properties of each region must be specified. A region may be designated as being electrically insulating, an electrode or electrically conducting. Insulating regions are modelled as perfect insulators, and no further properties are specified. Electrode regions are perfect conductors, and the voltage of each electrode region must be specified. This is done by assigning an electrode number to each electrode region and then assigning a voltage to each electrode number in another namelist (&INWELL). More than one electrode region may have the same electrode number. This allows electrodes with complex, non-rectangular, shapes to be modelled by several adjacent or overlapping electrode regions. For an electrically conducting region, the electrical conductivity must be specified. In EPEIOS, the only functional dependence of the electrical conductivity is on temperature. This dependence is modelled using the following cubic equation:

$$\sigma = C24 + \alpha_b\,(T - 24) + \alpha_c\,(T - 24)^2 + \alpha_d\,(T - 24)^3$$

where σ is the electrical conductivity in S/m, T is the temperature in Celsius, and $C24$, α_b , α_c , and α_d are user specified coefficients that are unique to each modelled medium. The values of these coefficients have been experimentally determined for many types of oilsand and shale. A region of anisotropic electrical conductivity may be modelled by inputting values for C24Y and C24Z if the principle axes of the conductivity tensor are parallel to the coordinate axes of the grid blocks.

The thermal properties of each region must also be specified. Regions may be designated as being thermally conducting, thermally insulating or at a constant temperature. If the region is a thermally conducting region, the volumetric heat capacity (THMM) and the thermal conductivity (THMK) must be specified. If a region is thermally insulating, no further thermal properties are needed. If the region is a constant temperature region, the temperature of the region must be specified. A region that is electrically conducting must also be specified as thermally conducting, or a program error will result. Electrically insulating or electrode regions may be specified as any type of thermal region. Any region that is electrically and/or thermally conducting must be at least 2*2*2 grid blocks in size or else a program error will occur.

MINX - ***Type: Integer, Default: None***
MINX specifies the minimum grid block number in the X direction for the region.

MAXX - ***Type: Integer, Default: None***
MAXX specifies the maximum grid block number in the X direction for the region.

MINY - ***Type: Integer, Default: None***
MINY specifies the minimum grid block number in the Y direction for the region.

MAXY - ***Type: Integer, Default: None***
MAXY specifies the maximum grid block number in the Y direction for the region.

MINZ	- ***Type: Integer, Default: None*** MINZ specifies the minimum grid block number in the Z direction for the region.
MAXZ	- ***Type: Integer, Default: None*** MAXZ specifies the maximum grid block number in the Z direction for the region.
ETYPE	- ***Type: Character, Default: None*** ETYPE specifies the electrical type of the region. ETYPE must be one of 'COND' indicating an electrical conducting region, 'ELEC' indicating an electrode region, or 'INSU' indicating an electrically insulating region.
ELENUM	- ***Type: Integer, Default: None*** If the region is an electrode, the electrode number is specified by ELENUM. Multiple regions may be used to form a single electrode by assigning the same electrode number to each region. The voltages of the electrodes are input in namelist &INWELL.
C24	- ***Type: Real, Default: None, Units:*** $S \cdot m^{-1}$ For electrically conducting regions, C24 specifies the electrical conductivity at 24°C. For the case of anisotropic electrical conductivity (when C24Y and C24Z are specified), C24 specifies the conductivity in the X direction.
C24Y	- ***Type: Real, Default: C24, Units:*** $S \cdot m^{-1}$ For regions of anisotropic electrical conductivity, the conductivity (at 24°C) in the Y direction is specified by C24Y. (For isotropic regions, C24Y does not need to be specified.)
C24Z	- ***Type: Real, Default: C24, Units:*** $S \cdot m^{-1}$ For regions of anisotropic electrical conductivity, the conductivity (at 24°C) in the Z direction is specified by C24Z. (For isotropic regions, C24Z does not need to be specified.)
ALPHAB	- ***Type: Real, Default: 0., Units:*** $°C^{-1}$ ALPHAB (α_b) specifies the coefficient for the term (T-24) in the cubic polynomial equation describing electrical conductivity as a function of temperature. Refer to the paragraphs at the start of this section.

ALPHAC	- ***Type: Real, Default: 0.0, Units:*** $^{\circ}C^{-1}$ ALPHAC (α_c) specifies the coefficient for the term $(T-24)^2$ in the cubic polynomial equation describing electrical conductivity as a function of temperature.
ALPHAD	- ***Type: Real, Default: 0.0, Units:*** $^{\circ}C^{-1}$ ALPHAD (α_d) specifies the coefficient for the term $(T-24)^3$ in the cubic polynomial equation describing electrical conductivity as a function of temperature.
TTYPE	- ***Type: Character, Default: None*** TTYPE specifies the thermal type of the region and must be one of 'INSU' indicating this is an thermally insulating region, 'CTEM' indicating this is a constant temperature region, or 'COND' indicating this is a thermally conducting region.
TEMPER	- ***Type: Real, Default: None, Units:*** $^{\circ}C$ TEMPER specifies the temperature of a constant temperature region (i.e. if TTYPE is 'CTEM', TEMPER must be specified.)
THMM	- ***Type: Real, Default: None, Units:*** $J \cdot m^{-3} \cdot K^{-1}$ For thermally conducting regions, THMM specifies the volumetric heat capacity of the region.
THMK	- ***Type: Real, Default: None, Units:*** $W \cdot m^{-1} \cdot K^{-1}$ For thermally conducting regions, THMK specifies the thermal conductivity of the region.
LPREG	- ***Type: Logical, Default: .TRUE.*** If LPREG is equal to .TRUE., all the input data for the region will be echoed to the printed output file. If LPREG is set to .FALSE. the printing of the data will be suppressed.

Namelist &INPUT3

The namelist &INPUT3 is used to specify the form of printed output, data storage output and data generated for graphing purposes.

The printed output contains such information as the input parameters, simulation time, CPU time used, current, voltage and power into the formation and energy balance calculations. In addition, when electrical heating is taking place, the electric potentials, heating rates and temperatures will be printed as specified. If no electrical heating is taking place, only the temperatures will be printed.

The three dimensional field data (temperature, heating rates, and potentials) are printed out as several X-Y planes at different Z values. There are two ways to specify the points on the X-Y planes and the Z values of the planes. The first way is to specify the number of X, Y, and Z points, and the minimum and maximum values of the points. The program then creates a regular grid between the maximum and minimum points and interpolates and prints out the grid values. The second way to specify the points is to set LUSPL to .TRUE. and explicitly specify each of the X, Y, and Z coordinates (in XP, YP, and ZP).

Printed output to a file will occur for the first and last timesteps, and may also be specified to occur every NSPO timesteps, or at explicitly specified times (PTIMES). Data storage output to a file, will occur for the first and last timesteps, every NSTO timesteps, and at times specified by PTIMES.

Graphical output for EPEIOS results is achieved by generating data files which can be imported into independent graphics packages. These data files provide values for simulation time, coordinates of the data points, potential (real and imaginary), heating rate and temperature.

Data representing a 2-D slice through the modelled configuration will be produced in tab delimited columns in ASCII X-Y format if LTAB is set equal to .TRUE.. If this format is chosen, the namelist variables NCUT, CUTS, and PLANE must also be specified. A new file is generated in a specific folder at each time that printed output is requested (specified by PTIMES and/or NSPO) and for each data slice that is requested. Each file is identified by a file name which is composed of the time step number and number of the data slice. The file name also includes the dimensions (columns x rows) of the data file. An example file name is as follows:

ts=10,XY2 (8 x 600)

Such a file name would identify a file that was generated at the tenth time step for the second data slice (specified in the variable CUTS) taken in the XY plane. The dimensions of this data file would be 8 columns by 600 rows. Data in these files is ordered such that the two dimensional coordinates are in the first two columns. The order in which these two coordinates appear is the same as the coordinate axis designation of the plane in which the data slice was taken. For the example file name given, the first column in the data file would be the x coordinate and the second column would be the y coordinate. The third column contains the third coordinate which represents the position of the two dimensional data slice. The fourth column contains the simulation time corresponding to the time step at which the data was generated.

Data in columns three and four are constant throughout each data file. Columns five through eight contain data values for real potential, imaginary potential, heating rate and temperature respectively. This tab delimited format is compatible with the commercial graphics packages Spyglass Transform, Systat and Cricket Graph.

Data files in Hierachical Data Format (HDF) can be generated with the ordinary Macintosh version of EPEIOS (EPEIOS.LS) if LHDF is set equal to .TRUE.. This format requires that the variable MAXCUBE also be specified. EPEIOS88 does not support HDF and use of HDF variables with this version of the program will result in an error at run time.

Data produced in HDF is representative of the entire 3-D modelled configuration. The order in which data appears in an HDF file is the simulation time, the x, y and z coordinates, real potential, imaginary potential, heating rate and temperature. Complete data sets are written for those simulation times at which printed output is requested (specified by PTIMES and/or NSPO). Data in HDF is written to a folder, with separate files being created for each simulation time at which data is requested. The HDF files are named with the time step number corresponding to the simulation time at which the file was written. The HDF format is compatible with the graphics program Spyglass Dicer which can produce 3-D color images.

NSTO	***- Type: Integer, Default: 10000*** If NSTO is greater than 99, no data storage will occur. If data storage is requested (NSTO < 99), data will written to a file for every NSTO timesteps.
NSPO	***- Type: Integer, Default: 10000*** Printed output and if requested, graphing data will be produced for each timestep number which is evenly divisible by NSPO.
PTIMES	***- Type: Real Array, Default: No Values, Units: seconds*** Printed output and if requested, storage output and graphing data will be produced for each simulation time specified in PTIMES. WARNING: The number of PTIMES specified must be less than the maximum number of regions allowed in the program version that you are running.
NPX	***- Type: Integer, Default: 10*** NPX specifies the number of X points in the print out grid for 3D field data (such as temperature).
NPY	***- Type: Integer, Default: 5*** NPY specifies the number of Y points in the print out grid for 3D field data.

NPZ	- ***Type: Integer, Default: 3*** NPZ specifies the number of Z points in the print out grid for 3D field data.
LUSPL	- ***Type: Logical, Default: .FALSE.*** If LUSPL is .FALSE., then the print out grid is a regularly spaced grid between the minimums and maximums specified by by XMIN, XMAX, YMIN, YMAX, ZMIN, and ZMAX. If LUSPL is .TRUE., the user must specify an irregularly spaced grid using XP, YP, and ZP.
XMIN	- ***Type: Real, Default: Middle of the first grid block, Units: meters*** This specifies the minimum X value for a regularly spaced print out grid.
XMAX	- ***Type: Real, Default: Middle of the last grid block, Units: meters*** This specifies the maximum X value for a regularly spaced print out grid.
YMIN	- ***Type: Real, Default: Middle of the last grid block, Units: meters*** This specifies the minimum Y value for a regularly spaced print out grid.
YMAX	- ***Type: Real, Default: Middle of the last grid block, Units: meters*** This specifies the maximum Y value for a regularly spaced print out grid.
ZMIN	- ***Type: Real, Default: Middle of the last grid block, Units: meters*** This specifies the minimum Z value for a regularly spaced print out grid.
ZMAX	- ***Type: Real, Default: Middle of the last grid block, Units: meters*** This specifies the maximum Z value for a regularly spaced print out grid.
XP	- ***Type: Real Array, Default: None, Units: meters*** XP is a vector specifying the X grid point values for 3D data print out. A total of NPX values must be specified if LUSPL is TRUE.
YP	- ***Type: Real Array, Default: None, Units: meters*** YP is a vector specifying the Y grid point values for 3D data print out. A total of NPY values must be specified if LUSPL is TRUE.
ZP	- ***Type: Real Array, Default: None, Units: meters*** ZP is a vector specifying the Z grid point values for 3D data print out. A total of NPZ values must be specified if LUSPL is TRUE.

LTAB - *Type: Logical, Default: .FALSE.*
If LTAB is set equal to .TRUE., data slices in tab delimited format will be generated at the times specified by PTIMES and NSPO. The orientation and position of each data slice is defined by the variables PLANE, CUTS, and NCUT which must also be specified.

PLANE - *Type: Character, Default: None*
Specifies the cartesian plane in which the values of the variables to be graphed are calculated. This variable may be specified as being one of 'XY','YZ', or 'XZ' representing the three possible planes.

CUTS - *Type: Real Array, Default: None, Units: meters*
The value of CUTS determines the position of the plane along the axis running perpendicular to it with respect to the origin. Several slices may be chosen up to a maximum of 5.

NCUT - *Type: Integer, Default: None*
NCUT specifies the number of different positions identified in CUTS. It will be an integer value in the range of 1 to 5.

LHDF - *Type: Logical, Default: .FALSE.*
If LHDF is set equal to .TRUE., a data set will be generated in HDF. MAXCUBE must also be specified.

MAXCUBE- *Type: Integer, Default: None*
For the data file in HDF, this variable defines the number of data points taken in the direction corresponding to the longest dimension of the configuration. The number of data points taken in other directions are scaled according to configuration dimensions.

Namelist &INWELL

The namelist &INWELL is used to specify the electrode voltages, wellbore resistances, type of heating, length of heating, and size of the timestep. Several &INWELL namelists may be included at the end of the input file and these will be read and executed in order. Thus, the heating of a formation by several different and successive excitation schemes may be simulated.

ELEVOL - *Type: Complex Array, Default: (0.0+j0.0), Units: Volts*
ELEVOL specifies the voltages of the electrodes in order (i.e. electrode

no. 1 is first, electrode no. 2 is second, etc.) Complex electrode voltages must be specified, even for single-phase problems (In this case, set the imaginary part to zero). For constant current and constant power problems the voltages will be scaled from the specified values.

WRESIS	- ***Type: Complex Array, Default: (0.0+j0.0), Units: Ohms*** WRESIS specifies the complex impedances of the well bores to which the corresponding electrodes are connected. This is used in calculations of the surface voltages and the electrical energy lost in the well bores.
L1PHAS	- ***Type: Logical, Default: .FALSE.*** If the problem being modelled is a single-phase problem (the imaginary part of all the electrode voltages are zero) then computational time and print out space may be saved by specifying L1PHAS as .TRUE..
NTYPE	- ***Type: Integer, Default: 1 (Constant Voltage)*** NTYPE is used to specify the electrical heating as Constant Voltage (NTYPE=1), Constant Current (NTYPE=2, and CCUR and NECUR must be specified), Constant Power (NTYPE=3, and CPOW must be specified) or no electrical heating (NTYPE=4).
CCUR	- ***Type: Real, Default: 0.0, Units: Amperes*** When modelling constant current heating, the magnitude of the current (CCUR) out of a specified electrode (NECUR) must be input.
NECUR	- ***Type: Integer, Default: None*** Specifies the electrode used when modelling constant current heating. Only one electrode may be specified as providing constant current.
CPOW	- ***Type: Real, Default: 0.0, Units: Watts*** Specifies the amount of power input into the formation when using constant power heating. Electrode voltages are automatically scaled to maintain constant power input.
TTIME	- ***Type: Real, Default: None, Units: seconds*** TTIME specifies the time to the end of the simulation, or the time at which the next namelist &INWELL is to be read. If TTIME is less than the current simulation time, the next &INWELL namelist will be read.
DTIME	- ***Type: Real, Default: TTIME/10. or previous value,Units: seconds*** DTIME specifies the length of the first timestep after the namelist is

read in. After the first timestep, the length of the timestep is automatically adjusted.

DTMAX - ***Type: Real, Default: 0.1***
DTMAX specifies the maximum relative change in temperature ($\Delta T / T$) that may occur in one timestep. If the temperature change is found to be greater than DTMAX, the timestep size is reduced and the temperature is recalculated.

DSIGMA - ***Type: Real, Default: 0.05***
DSIGMA specifies the maximum relative change in conductivity ($\Delta\sigma/\sigma$) which may occur in one timestep.

DTIMAX - ***Type: Real, Default: 1.E40, Units: Seconds***
DTIMAX specifies the maximum time step size.

ITMAX - ***Type: Integer, Default: 900***
ITMAX specifies the number of iterations of the potential solving point-wise successive overrelaxation routine which will be done before the program aborts if the convergence test has not been passed.

RESMAX - ***Type: Real, Default: 5.E-6***
RESMAX specifies the maximum relative residual of the finite difference equations for which the solution will be deemed to have converged.

MAXTEMP-***Type: Logical, Default: .False.***
If MAXTEMP is set to .TRUE., the simulation will stop when the maximum temperature within the heated formation reaches 100°C.

LRESET - ***Type: Logical, Default: .TRUE.***
LRESET specifies whether the potential array is to be cleared after reading in the namelist &INWELL. Setting this to .FALSE. may save calculation time in some types of problems (i.e. a restart of a stored run) but for the first &INWELL namelist of a new run this must be .TRUE..

EPEIOS Input File Template

On the following page is a copy of an EPEIOS input file. This file was used in a simulation of a physical model run.

```
&INPUT1
  LABEL1='EPEIOS Test Run. ET5 - Physical Model Case 8 ',
  LABEL2='Electrodes in both o.s. and o.b. and u.b. ',
  LABEL3='Constant current run, with storage',
  LABEL4='Geometry set up and entered by Allan Hiebert',
  MNSTEP=100,LRESTA=.FALSE.,NREC=1
&END
&INPUT2     NX=21,NY=34,NZ=6,
  DELTAX=.006,3*.01,6*.02,.028,5*.02,4*.01,.006,
  DELTAY=3*.1,.05,2*.025,8*.01,6*.02,8*.01,2*.025,4*.05,
  DELTAZ=6*0.0125,
  NREG=4,NELECT=2,TINIT=22.5,
&END
&REGION    MINX=1,MAXX=21,MINY=1,MAXY=34,MINZ=1,MAXZ=6,
  ETYPE='COND',ALPHAB=2.5E-2,C24=0.0079,
  TTYPE='COND',THMM=1.3E6,THMK=1.7,
&END
&REGION    MINX=1,MAXX=21,MINY=11,MAXY=24,MINZ=1,MAXZ=6,
  ETYPE='COND',ALPHAB=2.6E-2,C24=0.0014,
  TTYPE='COND',THMM=1.8E6,THMK=1.5,
&END
&REGION    MINX=1,MAXX=1,MINY=9,MAXY=13,MINZ=1,MAXZ=2,
  ETYPE='ELEC',ELENUM=1,
  TTYPE='COND',THMM=3.5E6,THMK=390.,
&END
&REGION    MINX=21,MAXX=21,MINY=22,MAXY=26,MINZ=1,MAXZ=2,
  ETYPE='ELEC',ELENUM=2,
  TTYPE='COND',THMM=3.5E6,THMK=390.,
&END
&INPUT3
  NSTO=10,NPX=4,NPY=5,NPZ=3,LUSPL=.TRUE.,
  XP=.0191,.0635,.1143,.1651,
  YP=.399,.4495,.4787,.542,.631,
  ZP=0.0,.0254,.0635,PTIMES=39.,607.,
&END
&INWELL
  TTIME=361.,NTYPE=2,ELEVOL=(-500.,0.),(500.,0.),
  WRESIS=(0.,0.),(0.,0.),
  DSIGMA=.10,ITMAX=800,DTMAX=.15,L1PHAS=.TRUE.,
  CCUR=.0825,NECUR=2,
&END
&INWELL
  TTIME=1253.,NTYPE=2,ELEVOL=(-500.,0.),(500.,0.),
  WRESIS=(0.,0.),(0.,0.),
  DSIGMA=.10,ITMAX=800,DTMAX=.15,L1PHAS=.TRUE.,
  CCUR=.0945,NECUR=2,LRESET=.FALSE.
&END
```

Compiling EPEIOS

Both variations of EPEIOS are designed to run in the MPW programming environment. The source code is broken down such that each subroutine and common block is contained in a separate file with the file name composed of the subroutine or common block name with an attached ending of '.F'. With the directory set for the EPEIOS folder, the source code is compiled and combined with the use of an MPW 'make file' (epeios.make or epeios88.make) which contains all the MPW commands necessary to generate the EPEIOS application. The make file is invoked with the MPW command 'Build EPEIOS'.

Only the files (subroutines) that have been altered since the last compilation time will be recompiled. However, changes to a common block will cause the subroutine(s) in which it is included to be recompiled on execution of the 'Build' command.

Since a given configuration is modelled as a collection of grid blocks divided into separate regions, the requisite dimensions of many of the array variables used in the program depend on the number of grid blocks used in each direction or on the number of regions defined. To help minimize the run time, provision has been made where by common block arrays can be initialized such that array size corresponds to the problem dimensions.

Arrays are dimensioned with the use of several MPW scripts. A script is a file containing a collection of MPW commands that are executed in succession. Each script is located in a separate file in the folder EPEIOS:SCRIPTS. These scripts must be identified to the MPW shell by including the full pathname of the folder containing the scripts, under the Set Commands line in the file MPW:UserStartup. In an effort to make the scripts compatible with other Macintosh systems, pathnames in the scripts have been replaced with the term {EPEIOSFOLDER}. This term must also be defined in the MPW file UserStartup in the following manner:

```
Set EPEIOSFOLDER "Pathname:EPEIOS:" ; Export EPEIOSFOLDER
```

where 'Pathname' is the full pathname to the EPEIOS folder (unique to each machine).

To change the dimensions of the array variables, enter the script 'Newcom' on the MPW worksheet. This will open the file EPEIOS:Scripts:Initcom. The values that are shown for xx, yy, zz and ww represent the respective maximums for the variables NX, NY, NZ and NREG in the namelist &INPUT2. The variables xx, yy and zz must have a value of at least 2 but no greater than 100. The variable ww also can not have a value greater than 100 but its minimum value should be 10. Although it is possible to initialize the arrays to accommodate a grid structure that is 100*100*100 grid blocks, the amount of RAM that is available for running the application will limit the size of the grid structure that can be handled (A grid structure that is 40*40*40 requires ~8MB of RAM). Double click on the number corresponding to the dimension that needs to be changed and then type in the new number.

Save and close the file Initcom when changes are complete and then invoke the script by entering 'initcom' on the worksheet. With initialization of the arrays complete, the EPEIOS application must be recompiled using the Build command.

Running EPEIOS.LS

With compilation completed, this application may be invoked either by entering 'EPEIOS' on the MPW worksheet with the directory set for the EPEIOS folder or by simply double clicking on the EPEIOS icon. Once invoked, an output screen will appear which shows the progress of the simulation.

All input files must be located in the directory EPEIOS:INPUT NAMELISTS:FOLDER: where 'FOLDER' can be any user specified name of a folder containing the input namelist file. The file is selected as input for the EPEIOS application via a standard dialog box which appears at run time. If the restart of a run using previously stored data is desired, the file containing the stored data must be located in the folder EPEIOS:RUNTIME.I/O with the name 'RESTART'.

All output files are directed to the folder EPEIOS:RUNTIME.I/O. The printed output appears in the file 'OUTPUT' and the stored data is written to the file 'STORAGE'. Graphing data in the tab delimited format is written to files in the folder 'TABFORM' and data in HDF is written to files in the folder 'HDFORM'. The 'Save as...' command should be used to save a file to a desired folder after a simulation is complete.

In order to avoid the mixing of data from two different simulations, the script 'Clearout' was created which, when invoked, deletes **all** output files from the folder EPEIOS:RUNTIME.I/O.

Running EPEIOS88

EPEIOS88 is invoked by first selecting the command 'Host88…' from the Build88 menu. This causes a dialog box to appear from which the user selects the application EPEIOS88 in the EPEIOS folder. After selection, a window will appear which displays information about the progress of the simulation.

EPEIOS88 reads the input namelist located in the file EPEIOS:RUNTIME.I/O: INPUT. Please note that the input file must end with a return character or EPEIOS88 will not be able to read the last namelist in the file. Also note that for this version of EPEIOS, complex variables in the input namelist must be grouped together with parenthesis in order to be read without error. If a restart is requested, EPEIOS88 will read the file EPEIOS:RUNTIME.I/O:RESTART.

Output files are written to the folder EPEIOS:RUNTIME.I/O. These are all the same as for EPEIOS.LS except that there is no folder for HDF data files (HDFFORM). All the output files in this folder may also be deleted by invoking the script 'Clearout' from the MPW worksheet.